UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ENGENHARIA GEOGRÁFICA, GEOFÍSICA E ENERGIA

Development of a multiple orientation solar radiation sensor

Luís Miguel Resende Martins

Mestrado Integrado em Engenharia da Energia e do Ambiente

Dissertação orientada por:
Miguel Centeno Brito

2018

Acknowledgements

I would like to thank my advisor, Miguel Centeno Brito for guiding and supporting me over the years. You have set an example of excellence as a researcher, mentor, instructor, and role model. I would equally like to thank Margarida Pedro, for the help and insight provided by EDP inovação, without which this project wouldn't have been the same.

I'd like to thank all the professors in the Department who taught me over the years. Every piece of what I achieve today and, in the future, has roots in your excellent work.

I'd like to thank Rodrigo Amaro e Silva and João Silva for all the time spent guiding me when they really should be doing something else.

I would like to thank my family for all the support given to me throughout my academic journey. Catarina, thank you for your undying spirit and persistence, every time I need to make an effort for something really hard, I ask myself for half your stubbornness. You are an example of true willpower. Filipa, I would like to thank you for your words of calm and sureness, and for being rock solid when everything else shook and tore apart. Most of what really matters still stands thanks to your reassuring love and advice. To my father, who never ceases to amaze me with the wisdom of the simple man. Thank you for everything I never went through thanks to your effort and dedication. I owe everything I am to you and you will always be my safe haven't. Dear mother, thank you for all the hard work. Your tough love and guidance made me the man I am today, and I couldn't be more thankful. Last but not least, I want to thank my aunt for opening her house to me when I needed. I keep you all in my heart forever.

To my longtime friends, Mário Cruz, Jéssica Marques, Bruno Serrano, Ana Marçal and Filipe Correia, who never failed me in love, laughter, advice and affection. You too share a special place in my heart. This is only the beginning.

To my friends in Oficina das Energias. Bernardo Tavares, you are the best partner in crime I could ask for. With you I learned that simple ideas and hard work can achieve true greatness. You showed me a new perspective of the world and you changed my life forever. I cannot say how much I am thankful to you. Together we did magic! Duarte, you are the best "slave" I could ever ask for and I only wish I could have taught you more. I trust you like I trust no one and I know you'll keep the magic going. I will forget none of you, André Borges, Rita Marouço, Raquel Castanho, Carole Matias, Carolina Menor, Beatriz Coelho, I wish you all the best in life!

To the friends from the hacking community, Tiago Rosa, Tiago Rorke and Maurício Martins, thank you for the shared knowledge and for always being available to discuss the most interesting topics in the making business.

To my love, Sofia. This endeavor was made easier by your effort, help and personal sacrifice. I only wish I could have met you sooner. Thank you for the sleepless nights, for the delicious food, for your patience and for your love. Thank you especially for listening to me talking permanently about things you know and care nothing about, knowing I needed to think aloud. I love you. Together we can achieve anything.

Abstract

The future implementation of urban photovoltaic power production is dependent on the furthering of the knowledge about spatiotemporal distribution of the solar resource. Current pyranometer technologies while accurate, still pose an obstacle to the development of city-wide mesh sensing grids due to their high cost and maintenance. Developing a low-cost radiation sensor is thus essential to the deployment of such mesh sending grids and consequently to the development and enhancement of information dependent energy systems like power grids and smart grids in the future. The rise of the internet of things provides the correct tools to tackle this issue. Using consumer level technology associated with prototyping tools we developed a wireless, self-powering, low-cost solar irradiance that estimates all three main components of solar light was developed. Global irradiation is collected and used to calculate direct and diffuse irradiance with an artificial neural network algorithm. The result is a low-cost sensor with no moving parts and substantially lower maintenance needs that effectively measures all three components of solar irradiance which are typically measured with devices far more expensive and elaborated.

Keywords:

Photovoltaics; Forecast; Sensor; Pyranometer; Direct Normal Irradiance; Diffuse Hemispherical Irradiance; Global Irradiance

Resumo alargado

O mercado energético está a mudar, e com o declínio dos combustíveis fósseis é a penetração cada vez maior das energias renováveis, que dita as tendências para o futuro. Estes novos tipos de recursos energéticos são, por natureza dispersos e variáveis com o tempo (Com grande ênfase na energia eólica e na energia solar fotovoltaica). Do ponto de vista do operador da rede surgem novos desafios tecnológicos todos os dias. Do ponto de vista do operador da rede surgem novos desafios tecnológicos todos os dias. Estes desafios prendem-se com uma maior instabilidade da rede devido a diversos fatores tecnológicos, sociais e económicos e deles resultam novos tipos de necessidade por parte dos consumidores e dos comercializadores e produtores de energia.

O abraçar dos recursos renováveis distribuídos como por exemplo a energia solar e eólica doméstica influencia diretamente o comportamento da rede elétrica. Com a proliferação da geração de energia fotovoltaica e eólica em meio urbano, os consumidores tornam-se produtores, surgem plataformas de agregação de produção e consumo e centrais virtuais de produção de energia. Com a proliferação da geração de energia fotovoltaica e eólica em meio urbano, os consumidores tornam-se produtores, surgem plataformas de agregação de produção e consumo e centrais virtuais de produção de energia. O surgimento destes modelos de consumo implica que a gestão da rede elétrica seja cada vez mais dependente da análise e previsão detalhadas (no tempo e no espaço) da variação do recurso solar e eólico disponível.

Este trabalho situa-se no âmbito da analiso do recurso solar fotovoltaico. A monitorização das condições meteorológicas é chave para a resolução deste problema, uma vez que por natureza, o recurso solar disponível pode ser encarado como uma variável meteorológica. Os presentes modelos meteorológicos baseiam-se em informação com uma amostragem de baixa densidade. Os presentes modelos meteorológicos baseiam-se em informação com uma amostragem de baixa densidade. Isto significa que, do ponto de vista do produtor de rede, certos eventos, como a passagem de nuvens e sombreamentos por prédios altos são fatores desconsiderados destas análises.

Em alternativa a este problema, surgem as soluções de mesh sensing, que tiram proveito de uma maior densidade de sensores para proporcionarem uma maior qualidade da informação dos modelos de previsão. Contudo, as tecnologias convencionais embora muito precisas, ainda são de custo de aquisição e manutenção demasiado elevados para que possam ser instaladas estações distribuídas em espaços urbanos. Este facto é ainda um obstáculo para o desenvolvimento de redes de mesh sensing a escala urbana que necessitam de pontos de medição com a maior densidade espacial possível.

O desenvolvimento de sensores de radiação mais baratos é essencial à implementação de tais redes e consequentemente ao desenvolvimento e aperfeiçoamento de sistemas de energia dependentes de informação, tais como as redes elétricas convencionais e as smart grids do futuro não muito longínquo. A proliferação dos sistemas de informação e dos smartphones, gerou sistemas consumidores e geradores de uma elevada quantidade de informação.

O aparecimento da internet das coisas traz-nos então as ferramentas necessárias para enfrentar este problema. Utilizando tecnologia de nível de consumidor associada com ferramentas de desenvolvimento de hardware e prototipagem, desenvolvemos um sensor de irradiância solar wireless autossuficiente e de baixo custo com base em eletrônica open-source que mede a irradiância global e estima as suas componentes direta e difusa.

Estas três componentes principais da luz solar são importantes para a compreensão e posterior modelação do recurso solar num certo local e consequentemente para a compreensão, modelação e previsão da produção solar fotovoltaica expectável nesse mesmo ponto. A medição de informação dispersa no espaço permite-nos fazer estimativas aproximadas com resoluções muito elevadas sobreo recurso solar num certo ponto e antecipar a passagem de nuvens e a ocorrência de sombreamentos nesses mesmos pontos.

Para a estimativa das várias componentes da radiação solar, foram utilizados dados do NREL (National Renewable Energies Laboratory), no Colorado (Estados Unidos da América). Os dados usados são relativos à posição do sol, á irradiância global, direta e difusa e às irradiâncias globais medidas por cinco piranómetros verticais virados para os pontos cardeais. Estes dados foram então usados para treinar uma rede neuronal de modo a que esta calculasse as diferentes componentes da radiação solar tendo apenas a informação da irradiância global de cada piranómetro e a posição do sol como input. Estes dados foram então usados para treinar uma rede neuronal de modo a que esta calculasse as diferentes componentes da radiação solar tendo apenas a informação da irradiância global de cada piranómetro e a posição do sol como input. O output da rede neuronal foi então testado e validado com os dados de irradiância difusa e direta do local. O produto é um modelo que estima com elevada precisão a irradiância direta e difusa a partir da direta.

O protótipo inicial é uma caixa de PVC de forma cúbica devidamente isolada e protegida contra os elementos. Em cada uma das faces do cubo foi instalado um painel solar fotovoltaico que serve de instrumento de medição do recurso solar e de unidade de produção de energia. Em cada uma das faces do cubo foi instalado um painel solar fotovoltaico que serve de instrumento de medição do recurso solar e de unidade de produção de energia. A medição do recurso solar tem por base a medição da corrente de curto circuito dos painéis solares e a utilização desta medida para a estimativa da irradiância global incidente em cada uma das seis faces do cubo. É importante salientar que apesar de o protótipo conseguir medir adequadamente todas as variáveis necessárias, ainda são necessários melhoramentos nos aspetos do isolamento contra os elementos e no circuito de alimentação, uma vez que, devido ao seu consumo elevado, o sistema ainda não é autossustentável e tem que operar ligado à rede.

Os dados obtidos pelo protótipo foram comparados com os dados do piranómetro medidos pela estação meteorológica da FCUL. Após essa comparação, é possível constatar que existe uma elevada correlação entre os dados da estação meteorológica e os dados da corrente de curto circuito do painel solar, o que nos permite substituir o piranómetro pelo painel solar, criando um sensor mais barato e com uma sensibilidade semelhante que representa uma alternativa mais barata face às técnicas convencionais de medição.

Concluiu-se que os dados do protótipo têm uma forte correlação com os dados da estação meteorológica. Assumindo que o dado de cada uma das faces do sensor equivale a um piranómetro convencional, a irradiância global de cada uma das faces é medida e usada para calcular a irradiância difusa e direta através do uso de um algoritmo de redes neuronais treinado com dados do NREL. O resultado é um sensor de baixo custo sem partes móveis e sem custos ou necessidades de manutenção, resultando numa opção de custo substancialmente mais baixo que mede com precisão as três componentes da irradiância solar que são normalmente medidas com equipamentos muito mais caros e complexos.

Futuramente, várias versões do produto podem ser desenhadas de modo a que haja uma maior sensibilidade, precisão e relevância nos dados obtidos para algumas aplicações. Como exemplo concreto

toma-se o número de painéis solares que pode e deve ser redimensionado, uma vez que os dados de três das seis faces não demonstram ser relevantes no estudo do recurso solar.

Palavras-chave:

Fotovoltaico; Previsão; Sensor; Piranómetro; Irradiância Direta Normal; Irradiância hemisférica difusa; Irradiância Global

Summary

Image Index

Tables Index

List of Abbreviations

ADC – Analog Digital Converter

ANN – Artificial Neural Network

BHI – Beam Hemispherical Irradiance

CS – Channel Select

DHI – Diffuse Hemispherical Irradiance

DIP – Double Inline Package

DNI – Direct Normal Irradiance

EDP – Energias de Portugal

GND – Ground

GPIO – General Purpose Input Output

GSM – Global System for Mobile Communications

HDMI – High Definition Media Input

I^2C – Inter-Integrated Circuit

IC – Integrated Circuit

IMU – Inertial Motion Unit

IOT – Internet of Things

I_{sc} – Short circuit current

MCU – Micro Controller unit

MISO – Master In Slave Out

ML – Machine Learning

MOSFET – Metal Oxide Field Effect Transistor

MOSI – Master In Slave Out

MUX – Multiplexer

NREL – National Renewable Energies Laboratory

NTC – Negative Temperature Correlation

PCB – Printed Circuit Board

PTC – Positive Temperature Correlation

PV - Photovoltaics

PWM – Pulse Width Modulation

RTC – Real Time Clock

RTD – Resistance Temperature Detector

RX - Receiver

SCK – Serial Clock

SDA – Serial Data

SPI – Seria Peripheral Interface

TTL – Transistor to Transistor Logic

TX - Transmitter

UART – Universal asynchronous receiver/transmitter

Vcc – Cathode Voltage

1. Introduction

1.1. Motivation

Solar resource measurement is of paramount importance for the renewable energy production sector. The readiness of the solar resource is one of the most viable alternatives to the current energy mix. The increased attention to the effects of pollution in dense population centers and greenhouse gases globally pushes forward the interest in PV. However, the solar resource is as useful as it is variable [Figure 1]. This bears severe consequences in the perspective of the power grid operator and it is often the most prominent reason why PV systems are used in the lower part of their potential range, if used at all.

The problem of forecasting available solar power stems from the problem of meteorological variables forecasting, since solar power production varies directly with the sky conditions above the installation site. This can pose a potential problem to the correct operation of power grids and discourage the pursuit of solar power as a viable alternative to the current energy mix.

The main issue of solar power production forecasting is that typical meteorology models have low spatial and temporal resolutions, due to computational and data availability constraints. Thus, this type of data is insufficient to recognize and predict the presence of clouds using current mathematical and statistical models. This derives from the nature of cloud formation which can be considerably smaller than the size of a single pixel of these models, thus, making cloud prediction and monitoring impossible without more in-depth models which carry a higher associated cost.

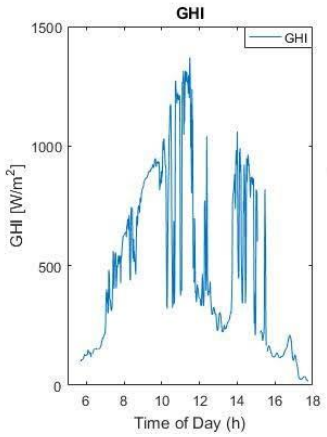


Figure 1 – Daily plot of GHI during a cloudy day.

Spatiotemporal models, using dispersed sensors in a wide area, have been shown to achieve higher accuracy for short-term cloud cover forecasting. Low-cost sensor solutions can contribute to a higher number of sensors in different locations, improving any models that rely on accurate dispersed solar radiation measurement. In using solar cells as a radiation sensor, errors are less relevant for PV application because measurements are made using the same technology, hence with the same spectral and angular response.

The rise of the *maker* movement in the last 10 years has caused a renewed interest in remote sensing solutions as well as in physical computing and robotics. This lowers the cost of instrumentation, enabling the design of capable, easy to make, low cost sensors, which can be the precursor for large city-wide sensor networks. The use of widespread internet communications on a large number of devices equipped with various sensors can also lead to a larger number of datasets and variables available for study. With the increased use and demand for this kind of connectivity comes a greater effort in financing and developing

widespread sensor solutions that use open source and open data frameworks. This demand in interconnectivity comes also from the integration of data collection across multiple platforms. The massive spread of IoT platforms like the Arduino and Raspberry pi are changing the technological landscape toward cheaper and more inventive solutions to different sectors.

User information and big data is central to developing integrated city-wide energy and transportation systems and there is a great demand for data collection, sensing and data processing platforms. All these factors combined imply that we are facing towards a future where data collection is key.

The solar resource is distributed evenly across the world, meaning that it can be produced virtually anywhere. Better information about the solar resource may imply a use of mesh sensing networks across large production sites. The use of smart-grid integrated PV in an urban landscape is one of the sectors that mostly benefits the implementation of low-cost sensor mesh networks since they can be integrated in the automatic decision process that characterizes smart grids. The use of wide area sensor arrays and new data processing techniques, such as artificial neural networks, genetic algorithms and artificial intelligence is the future of the energy sector.

By measuring multi-directional irradiance, one can create models to make building integrated photovoltaics a more predictable energy source. The dispersed measuring of solar potential allied to the widespread use of smart grids and artificial intelligence is a key factor for future cities.

1.2. Objectives

The purpose of this project is to develop and test a network of low-cost solar radiation sensors that can be used for improved spatiotemporal forecasting and thus enabling the implementation of better power grid management strategies. The system to be developed ought to be able to estimate the diffuse component of the solar radiation from the global component using software.

The sensor [Figure 2] is comprised of six solar panels in a cube frame. Each panel has a thermistor for temperature correction. There is an IMU mounted inside the cube that provides information on its tilt and orientation. The device communicates data from all sensors via GSM to a database on the cloud.

The data will be compared to data from a pyranometer placed next to the cube to assert whether the short circuit current from a solar panel is a suitable substitute for a pyranometer. Furthermore, data from several pyranometers with different orientations will be used to model different solar radiation components using Artificial neural networks.

An Artificial neural network model will be taught using varied data relevant to solar resource monitoring. When the learning process is completed, the ANN's predictions will be evaluated in order to determine if this is an accurate solar forecast model. The data acquired using the network of solar radiation sensors will be used for spatiotemporal regression forecasting. After the ANN concept is proven, the multi-directional sensor will be prototyped.

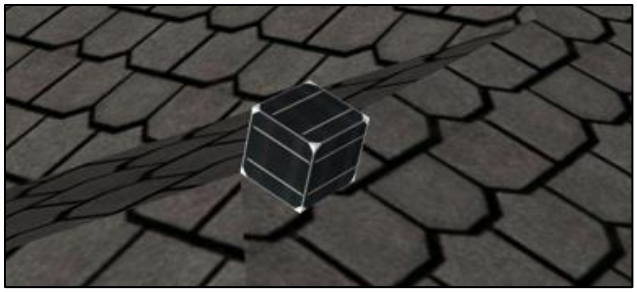


Figure 2 - Artistic rendering of the final product.

1.3. Overview

In order to better comprehend the scope of what is available to evaluate the solar resource, first one must assess the current technologies and techniques used in the industry. This will be done in chapter 2 where we will discuss different approaches on measuring and estimating the different components of solar irradiance. After this analysis is completed, in chapter 3 we will discuss the development of the multiple orientation solar radiation sensor itself. This approach will comprise both the hardware and the firmware required for the proper function of the sensor system, as well as notes on the construction of the sensor enclosure. The experimental results obtained from the sensor tests will be discussed in chapter 4. In chapter 5 we will assess artificial neural networks as a means of converting multiple orientation irradiance data into different solar irradiance components. Finally, in chapter 6 we will discuss the importance of this data and recommend future adaptations for the sensor's hardware and software, as well as new approaches for the data treatment.

2. State of the Art

2.1. Measurement of solar radiation components

The current technology for solar variable measuring is based on two regular devices: the pyranometer [Figure 4] and the pyrheliometer [Figure 3]. Pyranometers measure global irradiance [W/m2] using the thermopile principle. Inside a clear glass dome, there are two surfaces with different reflective indexes and high emissivity. When irradiation from the surroundings hits these surfaces, one of them heats up relatively to the other, creating a voltage difference that can be measured.

Through calibration curves, one can then convert directly from [mV] to [W/m^2]. This apparatus, however, cannot measure diffuse horizontal irradiation (DHI) or beam horizontal irradiation (BHI) without shades that either block the sun or block everything but the sun. The latter example is what is called a pyrheliometer. Both the pyrheliometer and the shaded pyranometer imply some sort of solar tracking to function correctly, having more moving parts, and they are therefore relatively expensive to install, operate and maintain.

Figure 3 - Shaded pyranometer [1]

One alternative to this is the SPN1 Sunshine Pyranometer, which uses a grid of sensors and static shaders to infer DHI and BHI mathematically from the GHI measurements. This alternative is of easy installation and maintenance but has an even higher cost then regular pyranometers. ("SPN1 Sunshine Sensor in Photovoltaics," 2014) Furthermore, it assumes that the diffuse irradiation is isotropic which is an inaccurate assumption in many occasions and applications.

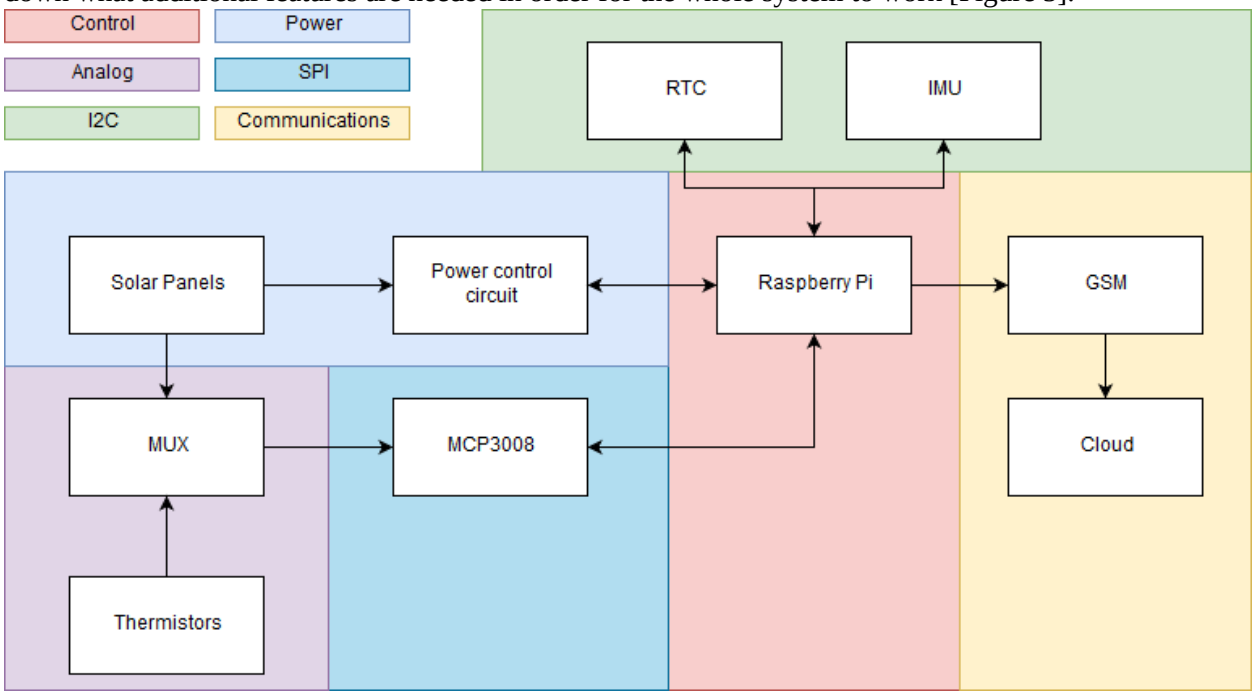
Figure 4 - SPN1 Sunshine pyranometer [2]

2.2. Solar radiation on surfaces with different tilts and orientations

The approach to be explored for the estimation of the different components of the solar radiation is to note that since surfaces with different tilts and orientations 'see' different parts of the sky and therefore their combination, they might yield indirect information on DNI, DHI and GHI. Hence, it is appropriate to review the irradiation that falls on a surface with arbitrary position. This can be done using the Perez model (Perez et al, 1990). It is not practical to analytically relate the irradiation incident on surfaces with different orientations (and horizontal surface) to extract the DHI and DNI and therefore we shall approach the problem with machine learning techniques, in particular Artificial Neural Networks.

3. Development of a multiple orientation solar radiation sensor

The prototype is to be developed through an iterative process, starting with is main features and working down what additional features are needed in order for the whole system to work [Figure 5].



3.1. Figure 5 - System Overview**Requirements**

The initially specified requirements for the multi-directional solar radiation sensor are as follows:

- Self-powering
- Raspberry Pi
- GSM communications
- 1 sample per minute
- Long range wireless communication
- Solar panel temperature measurement
- Solar panel short-circuit measuring
- X and Y tilt and orientation

3.2. Micro-Controller Unit

3.2.1. Raspberry Pi

To implement the project, one of the requirements was the use of the Raspberry Pi as the MCU. This hardware is more easily integrated onto a Linux operating system via USB which will be required to function with the GSM module. For this reason, the Raspberry pi 3 model B [Figure 6] was chosen to be the definitive MCU even though its energy demand is much higher. For our GSM module we are using a HUAUEI E-173 GSM dongle [Figure 7].

Figure 6 - Raspberry Pi model 3 B [3]

Figure 7 - Huawei GSM dongle [4]

3.3. Analog to digital conversion

The system in its whole has need for a total of 12 analog signal inputs, 6 for the temperature readings and 6 for the analog short circuit current measurements, but the Raspberry Pi is unable to measure analog values. To get around this problem we will use a CD74 HC4067 multiplexer, along with an ADC (analog to digital converter). To convert analog values into digital values we implemented the use of the MCP3004 IC [Figure 8]. It is a 10-bit analog to digital converter (ADC) with four channels and SPI (Serial Peripheral Interface) communication. The working principle for the MCP3004 is based on a clock signal (generated by the MCU) and the analog signal being fed onto an internal comparator IC. The result of the binary comparison is a 10-bit PWM (Pulse Width Modulation) [Figure 9] signal that the MCU can read and convert into a variable (see Figure 18). The first step in this conversion is to calculate the voltage associated with each Analog Direct Current value. Since we are using a 10-bit ADC what we will receive from the analog measurement is a value between 0 and 1023, with 0 corresponding to 0[V] and 1023 with 5[V]. The resolution associated with this measurement is calculated in [equation 1].

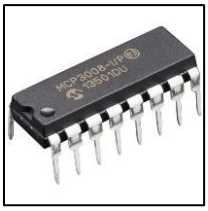


Figure 8 - MPC3008 [6]

$$\Delta V_{min} = \frac{5[V]}{1023[bit]} = 4.88\ \left[\frac{mV}{bit}\right] \quad (1)$$

Each wave from the PWM signal is divided into 1024 bits, each bit measure corresponds to the period of the MCU's internal clock. For the Raspberry Pi 3 model B, the clock frequency is 800[MHz], this means that the time required to make one measurement with the ADC is:

$$t_m = \frac{1024[bit]}{800[MHz]} = 1.28[\mu s] \quad (2)$$

This time is small enough to allow rapid sampling of all the analog inputs, much faster than the one-minute period time required for the sensor. The SPI device used is the MCP3008 ADC (analog to digital converter) which will bring the analog signals from the various sensors and relay them to the MCU in digital form. The device supports up to 8 analog channels, but since we are already using a 16-channel multiplexer, we will use only one of the 8 channels. The device is powered by the standard TTL voltage levels provided by the Raspberry pi. The different channels can be read through a specific SPI interface python library for the MCP3008 and MCP3004 ICs.

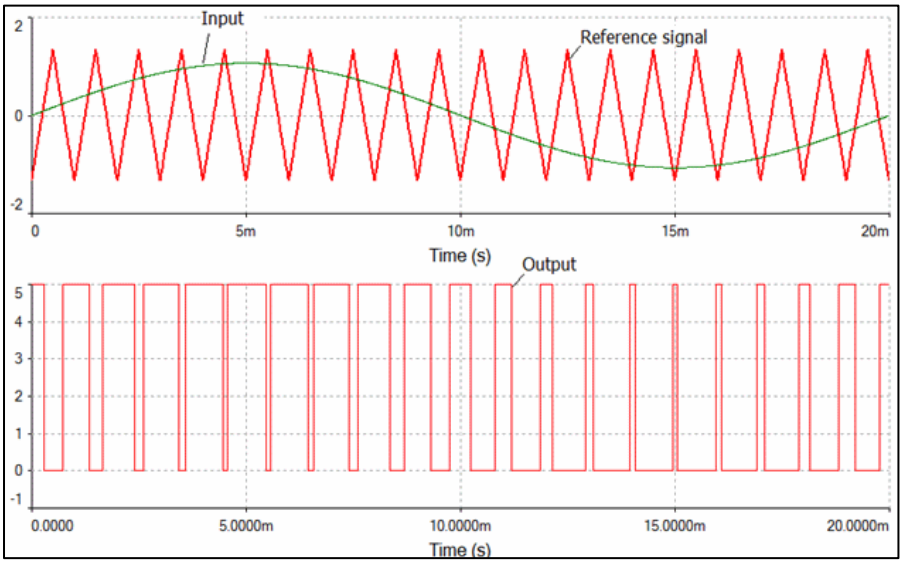


Figure 9 - Analog to Digital conversion process [5]

3.4. Serial Peripheral Interface

SPI stands for serial peripheral interface. It is a communication protocol to send and receive data between microcontrollers and peripheral sensors and a popular alternative to standard Serial communication. The standard Serial communication relies on two communication lines, one for receiving data (RX) and one for transmitting data (TX) (Figure 10). Serial communication is an asynchronous protocol, since the clocks of the devices communicating are not electronically connected. This can be a problem when two systems are communicating at different clock rates, resulting in failed or faulty communication. This problem is worked around by placing start and stop bits at the start of each sent byte to help the receiver synchronize with the receiver. Even so, both sides must agree on the transmission speed (often called baud rate) in advance. The baud rate is usually 9600 bits per second, which is very slow compared to the 1.2[GHz] from the Raspberry Pi.

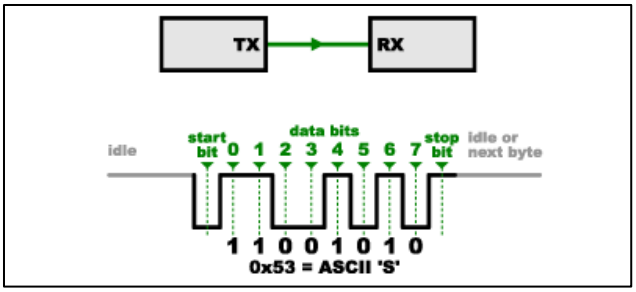


Figure 10 - Asynchronous Serial Protocol [5]

The asynchronous Serial protocol is slower than other protocols since it has to send an extra 2 bits per byte, resulting in 20% slower communication. The complex hardware needed to send and receive data correctly is also one of the drawbacks of this technology.

SPI on the other hand is a synchronous communication protocol (Figure 11) and works in a different manner. The first major difference is the shared clock line (SCK from Serial Clock), that ensures that all the devices communicate at the same rate, rendering extra communication hardware UART (Universal Asynchronous Receiver-Transmitter) needless and removing the start and stop bytes, thus increasing the communication speed.

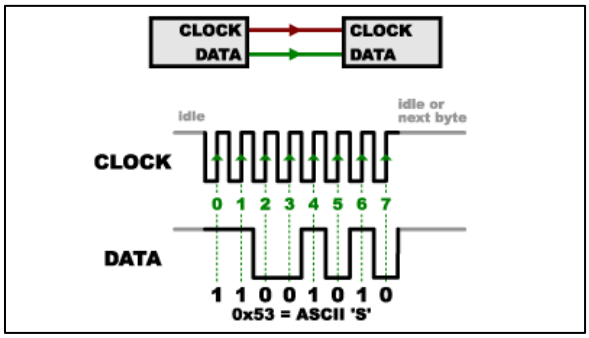


Figure 11 - Synchronous SPI protocol [5]

3.5. Receiving Data

So far, we have seen how the SPI keeps all devices in synchronous operation, but we have not yet explored how the protocol communicates in both directions with the devices. The devices are organized in a two-level hierarchy. The Master device is the one generating the clock signal for all the other devices while the slave devices are the peripherals. There can only be one Master device, whereas the number of slave devices is mostly dictated by the needs of the user in terms of pin usage (since each slave requires one chip select line) and communication speed. When sending data from the master to the slave, the MOSI (Master Out Slave In) data line is used. In order to receive data from the slave, the MISO (Master In Slave Out) line is used (Figure 12).

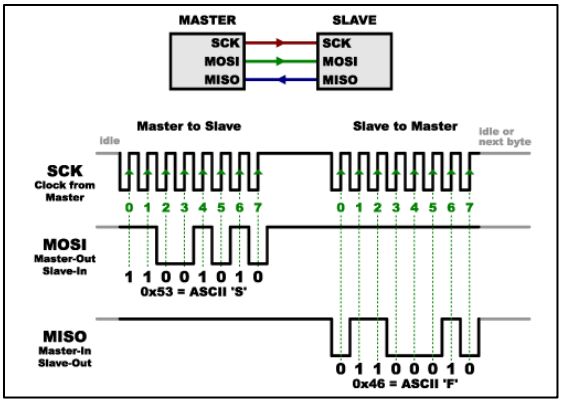


Figure 12 – Bi-directional SPI [5]

Due to its architecture, the SPI protocol can request data from one slave while reading data from the other, this means that it can send and receive data at the same time between two different slave devices further improving communication speed.

Having multiple devices sharing communication lines means that there may be some overlap in the communication, resulting in failed readings. To select which of the slave devices we are communicating with, there is another line called chip select (CS). Each slave is connected to the Master using a different CS line, and the master uses these lines to "awaken" the device it wants to write to/read from. The full protocol can be represented by the following schematic (Figure 13):

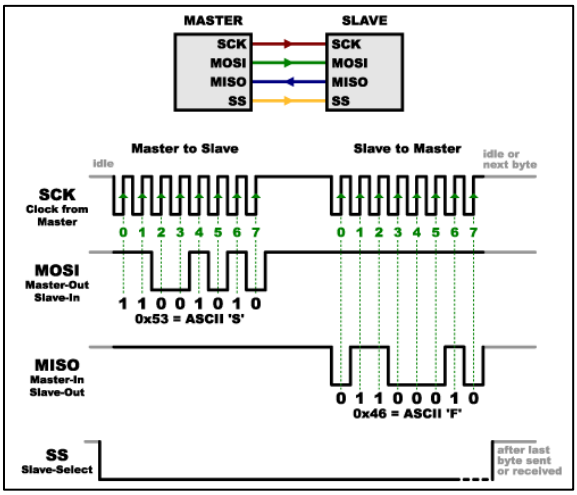


Figure 13 - SPI Channel Select [5]

One of the major limitations of SPI connections is the high number of lines required (4 is the minimum). The fact that an extra line from the device to the MCU is required for every additional device only furthers this problem. This is challenging because most MCU's have a limited amount of GPIO pins that could be better suited for use in other functions as inputs or outputs (FIG 14).

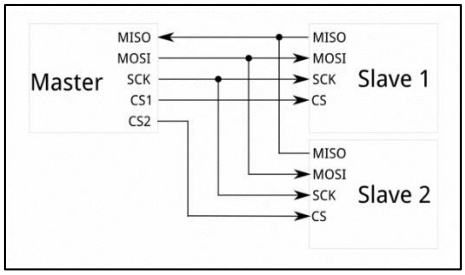


Figure 14 - Parallel SPI protocol [5]

Whilst some simple solutions like adding shift registers and multiplexers to the CS lines in order to use less GPIO pins, this always comes with a compromise in communication speed. In this device, only one SPI master-slave pair is used and as a consequence, this is not an issue. SPI connection is good for high speed date rates and simplifies the hardware needed for communicating making it a standard choice in many sensor applications.

3.6. Temperature

In order to measure temperatures, a simple voltage divider with an NTC (Negative Temperature coefficient) thermistor was used. The NTC is a temperature-dependent resistor that decreases its resistance with temperature (Figure 15). For our specific NTC, the resistor used was of 1500[ohm]. In relation to other temperature dependent resistors like resistance temperature detectors (RTD) the NTC has a steeper curve resulting in a larger variation and a narrower operation temperature gap from -55[ºC] to 220[ºC].

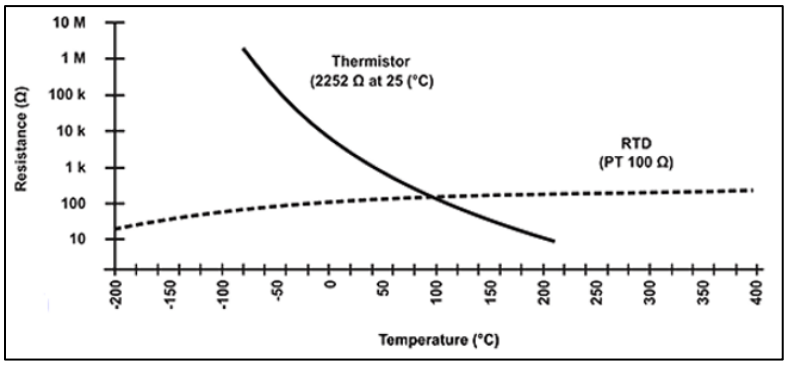


Figure 15 - Thermistor and RTD sensitivity comparison [6]

This is useful to us since we are restricted to a 1024-bit measurement, allowing us to maximize sensibility. Since the thermistors are mounted in the back of the solar panels, they are not expected to reach temperatures outside the NTCs temperature gap, therefore, these are the components used. For this project the chosen NTCs have a resistance of 1500[Ohm] (at room temperature of 25ºC) and a B value of 3950 as seen on the datasheet. The temperature measuring circuit is as follows (Figure 16).

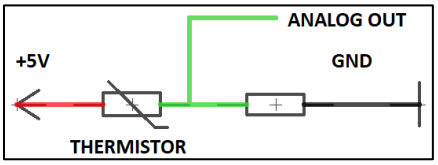


Figure 16 - Thermistor circuit schematic

When measuring environmental variables in digital systems, it is important to know how the voltage variations (which is what we are really measuring) correlate to the variable being measured by a certain sensor. For our thermistors the measured voltage indicates a change in resistance that can then be converted into a temperature value.

Using Ohm's law, we calculate the resistance of the thermistor at any given time so that we can use it in the Steinhart-hart correlation. We will refer to the thermistor's resistance value as R, the fixed series resistor as R_0, the 5[V] input voltage as V_{cc}, the reference voltage as A_{REF} and the measured voltage as V_m. The Thermistor's resistance is given by:

$$R = \frac{R_0 * (ADC - 1)}{(1023)} \tag{3}$$

where ADC is the voltage across the thermistor converted into bits. Now that we know the thermistor's resistance, we can now use the Steinhart-Hart correlation.

$$\frac{1}{T} = A + B * ln(R) + C * (\ln(R))^3 \tag{4}$$

The parameters A, B and C characterize the Thermistor curve and can be found in the thermistor's datasheet. Although this approach is more accurate, it requires access to all parameters in order to be implemented and the datasheet for the part in use included only the B value. For this we will use the Simplified B-value correlation (Equation 5)

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} * \ln\left(\frac{R}{R_0}\right) \Leftrightarrow T[ºC] = \left(\frac{1}{\frac{1}{T_o} + \frac{\ln\left(\frac{R}{R_o}\right)}{B}}\right) - 273.15[K] \tag{5}$$

where T_0 is 297[K] (25ºC), B is the B-value = 3950 (see table XX) and R_0 is the series resistor value. After these calculations the temperature value can then be used.

3.7. Short-circuit current

For measuring short circuit current, a shunt resistor with value R_s=10 Ω connected at the output of each solar panel between the positive and negative terminals (Figure 17). This resistor is of a small value and it will cause the voltage across the terminals of the solar panel to drop considerably. A voltage probe is then connected to the positive terminal of the solar panel the values measured are the potential difference (ΔV) across the test resistor. By applying Ohm's law (Equation 6), one can calculate the short circuit current (I_{sc}).

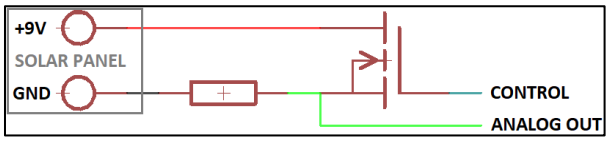


Figure 17 - short circuit current measuring circuit

$$I_{sc} = \frac{\Delta V}{R} \tag{6}$$

The resolution of the MCU's analog measurements is of 10 bits (1024 values) and its measuring range is 5 [V]. The resolution of the current sensor is given by equation 7:

$$\Delta I_{sc_{min}} = \frac{10.0[V]}{10[\Omega] * 1023[bit]} \approx 0{,}98\left[\frac{mA}{bit}\right] \tag{7}$$

The latter circuit will provide an analog signal between 0[V] and 10.5 [V]. These values are higher than the MCU's TTL levels and would damage it. In order to prevent damage, a voltage divider [Figure 18] is applied to reduce the voltage in half. The voltage divider has a resistance of 2000ohm, much higher than the test resistor and it will not interfere in the readings. Adding the voltage divider has beneficial effects on the resolution of the measurement increasing it by a factor of 2.

$$\Delta I_{sc_{min}} = \frac{5[V]}{10[\Omega] * 1023[bit]} \approx 0{,}400 \left[\frac{mA}{bit}\right] \quad (8)$$

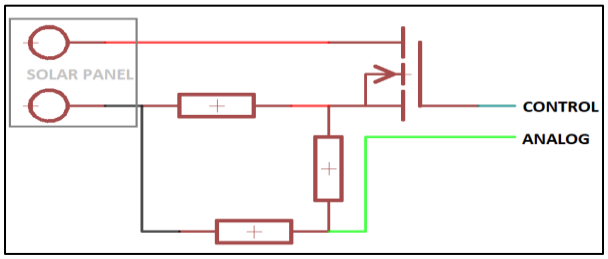


Figure 18 - Voltage divider in test resistor output

The combined circuit with the short circuit measurement test resistors and the voltage regulation IC is shown below (Figure 19):

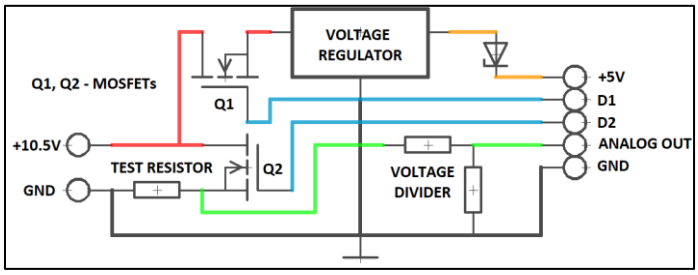


Figure 19 - Complete circuit

3.8. I2C Protocol

I²C requires two wires, like asynchronous serial, but those two wires can support up to 1008 slave devices. Also, unlike SPI, I²C can support a multi-master system, allowing more than one master to communicate with all devices on the bus (although the master devices can't talk to each other over the bus and must take turns using the bus lines)[Figure 20].

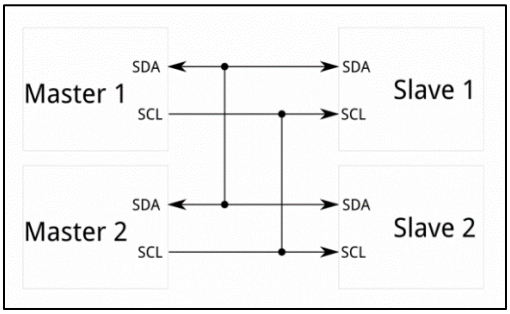


Figure 20 - I2C parallel protocol [5]

Data rates fall between asynchronous serial and SPI; most I²C devices can communicate at 100kHz or 400kHz. There is some overhead with I²C; for every 8 bits of data to be sent, one extra bit of meta data must be transmitted. The hardware required to implement I²C is more complex than SPI, but less than asynchronous serial communication. It can be fairly trivially implemented in software. In our project we have two I²C slave devices with the Raspberry pi MCU as the master device in addition to the SPI. I²C communication consists of two lines only. One for clock synchronizing and another for data transmission, regardless of the number of slaves and/or masters. The communications are carried on as shown in figure 21. A start byte tells all devices to listen for the slave address. Each slave device has a different address dictated by the hardware and only the device correspondent to the 7-bit address given by the master will awake. Next, the master sends a request bit (write or read) followed by a response from the slave consisting of an ACK (acknowledgement) bit. If the last byte is a 0 this means that the communication was not read properly and must be sent again, if it is a 1 this means the slave was selected successfully and that the information bytes can now be sent or read, depending on whether a read or write bit was sent.

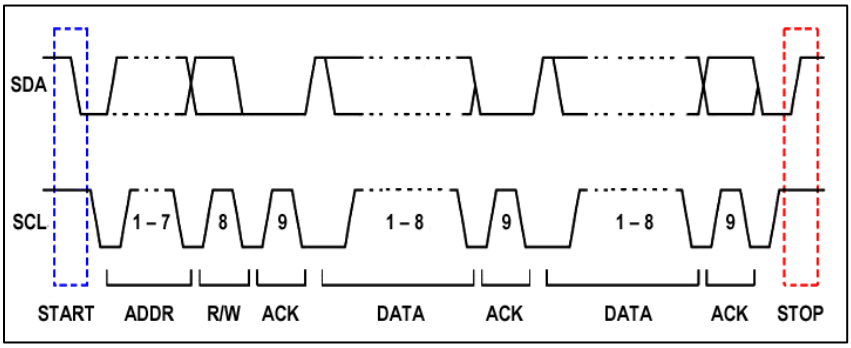


Figure 21 - I2C protocol communication [5]

3.9. Timekeeping

The first I2C device is an RTC (Real-time Clock) unit [Figure 22], that ensures that even when the system shuts down, it does not lose track of time. It also enables the use of timestamps associated with each data point that will serve as a trigger for the Edge Server application to know when to retrieve data. The default address of this device is 0x68. The Raspberry pi is not equipped with an RTC unit and relies on an internet connection to tell time and date reliably. This function is called fake hardware clock. The I2C RTC unit replaces this timekeeping system so that it works with no internet connection in situations where supply power is not continuous.

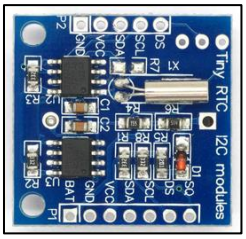


Figure 22 - Real Time Clock [7]

3.10. Accelerometer and Gyroscope

The second I2C device used is a 6 degree of freedom IMU (inertial motion unit) [Figure 23] with triple-axis accelerometer and gyroscope functions called MPU6050. The purpose of this component is to measure

the tilt of the surface in which the cube is placed. The default address of this device is the same as the default address for the real time clock. This means that unless one of the addresses is changed, these two hardware components will conflict with each other. Fortunately, the MPU board has an integrated address switch jumper that allows the user to change the address to 0x69, ending the compatibility issues.

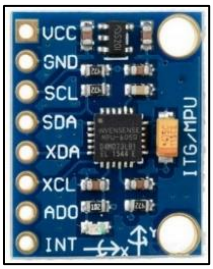


Figure 23 - Inertial Motion Unit [8]

3.11. Analog Multiplexing

The final output of the current and temperature measuring circuits is an array of 12 analog signals ranged between 0 and 5V. In order to use the MCP3004 chip for analog to digital conversion, one must first reduce the output of all the analog sensors to a single analog output line. To achieve this, a 74HC4067 multiplexer is used [Figure 24]. It features 16 to 1 channel multiplexing and is driven by raw GPIO signals in 4-bit binary code.

Figure 24 - Analog Multiplexer [9]

3.12. Mode switching

3.12.1. Using MOSFETs

The self-powering circuit was designed through an iterative process. The first step was to ensure the two different operating modes for this part of the circuit: powering the device and measuring the short circuit current. These may seem simple operations, but due to the nature of the short circuit current measurement, each solar panel and respective test resistor must be isolated from the rest of the circuit in order for accurate measurement to be possible. To achieve this, the following circuit was developed using IRF-520 Metal Oxide Field Effect Transistors (MOSFET) which is an N-type Power MOSFET. The working principle is as follows: apply a positive voltage to the gate pin and current will flow from the drain pin to the source pin (Figure 25). Using this, one would need a total of 12 MOSFETs for the total circuit (two for each solar panel), one to turn on the test resistor and another to turn off the powering circuit.

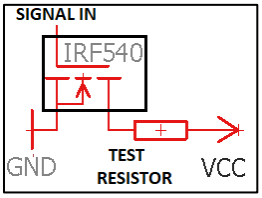


Figure 25 - MOSFET usage example

3.12.2. Adding a Hex-Inverter

Each measuring/powering circuit has two digital inputs and one analog output. As for the digital inputs, there is a total of 12. This poses a problem, because most MCUs do not have twelve available GPIO pins to use for this task. In order to circumvent this issue, an inverter array can be used as seen in (Figure 26). This way, only one of the GPIO outputs is used to change the operating mode across all the sensor circuits.

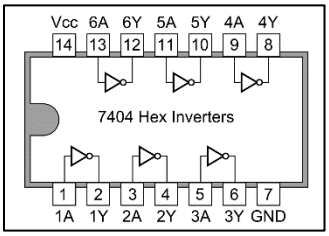


Figure 26 - 7404 Hex inverter [10]

3.12.3. Implementing a Voltage Doubler

To have correct measurements, we need to make sure the voltage at the gate (V_G) is close to 10V so that the resistance across the MOSFET drops to the value stated in the product datasheet (V_GS = 0,27 ohm). At this resistance value, voltage drop on the MOSFET is minimal and has an effect of 2% on out measured values:

$$V_{V1} = 9{,}5[V] - \left(0{,}27[\Omega] * \frac{9{,}5[V]}{0{,}27[\Omega] + 10[\Omega\,]}\right) = 9{,}25[V]$$

One way of achieving this is by applying a voltage doubler circuit to the digital signal input as shown in figure 28.

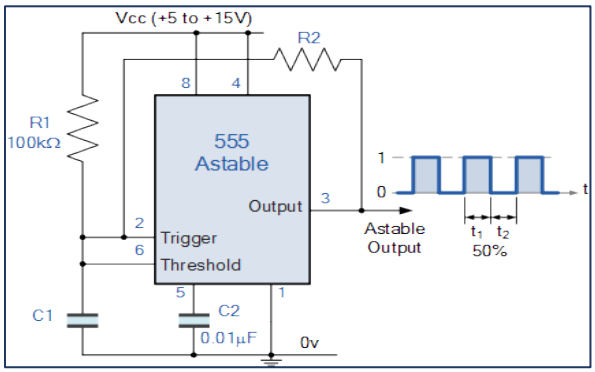


Figure 27 - Heaviside function generator circuit [11]

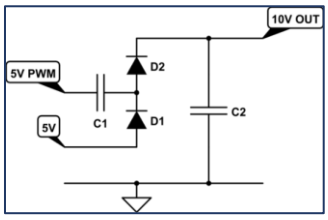


Figure 28 - Voltage Doubler [12]

This would imply that a GPIO pin would be used for generating the oscillating Heaviside function (PWM signal) or that an oscillator circuit (Figure 27) would be created to generate the PWM signal in order for the voltage doubler to function properly. The values measured in the laboratory for our voltage doubler output were of 8.56[V]. However, the dissipated heat on the MOSFET integrated circuit would be considerably high resulting in the destruction of the integrated circuit.

$$P_D = (9{,}25[V]) * \left(\frac{9{,}5[V]}{0{,}27[\Omega] + 10[\Omega\,]}\right) = 8.55[W] \tag{9}$$

3.12.4. Relays

Due to this fact, SPST switch relays were chosen to perform the mode switching operation. To ensure the desired functioning, each mode switching was connected to a relay. The powering half of the circuit is connected between the NC (normally closed) terminal and GND. The test resistor is mounted between the NO (normally open) terminal and GND (Figure 29). When no signal is provided, the test resistor is physically disconnected from the circuit and when the TTL signal is provided to the relay, only the test resistor is connected to the solar panels. This lets us make the measurements without any residual resistance from active components such as transistors and MOSFETs it also enables us to discard the voltage doubler circuit and the digital multiplexing circuit. However, there is a disadvantage to this approach. Though the relay is more practical, it still relies on mobile parts and will wear out in time, relays are also bigger in size than transistors and MOSFETs. The final circuit (Figure 30) is composed of six mode switching units composed by a relay, a test resistor, a voltage regulator and a solar panel.

Figure 29 - Switching Relay [13]

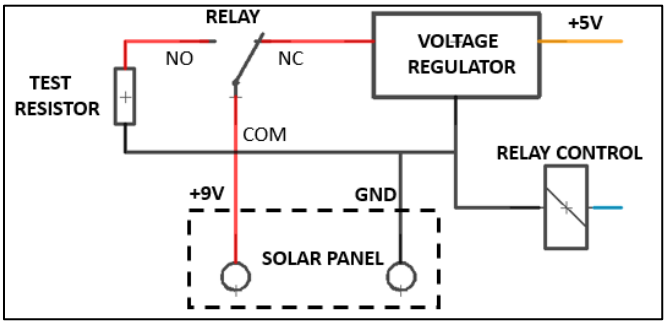


Figure 30 - Final Switching circuit

The code for the mode switching is very simple and uses one GPIO pin to switch states among all the solar panels. The modes can be changed using the “switch (MODE)” command. The two possible modes are “charging” and “measuring”.

3.13. Self-Charging

In order to power the device, first we need to consider the typical current draw values (I) and the operating voltages (V) that are standard for each of its components. In the preliminary stages of this project there was a great degree of uncertainty in the amount of power required, due to the experimental nature of developing the sensor. Regarding this, the entirety of the self-powering system was oversized in order to have greater plasticity when choosing the sensors, the MCU and the communications systems. The power flow on the device is controlled by voltage regulators in the sensor and by the charge controller in the powerbank (Figure 31). The raspberry pi drives all the main components.

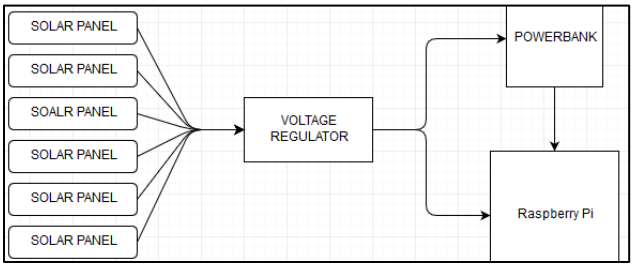


Figure 31 – Self-Charging operation diagram

The final power consumption can be measured using a controllable power supply in order to optimize the panel sizes and specifications after the project is concluded. This procedure is reported in section 4.1 of this work.

3.13.1. Solar Panels

The selected solar panels are the (STAR SOLAR CNC 165x165-9 model) (Figure 32). The I_{mp} and V_{mp} values determine the maximum power the solar panel can output. In our case, these values are of $500[mA]$ and $9[V]$ respectively and they constitute a total $4.5[W]$ of power (when at the maximum power point). Due to the cubic geometry of the device, where each solar panel is pointing towards a different direction, the total power produced is not equal to six times the nominal power of one panel. This is one of the reasons why such high-power panels were chosen for the self-charging process. Even in overcast days there should be enough power in the sum of all the solar panels for the device to work properly throughout the day.

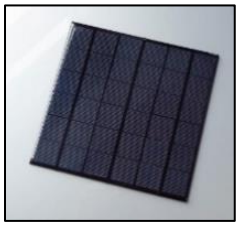


Figure 32 -STAR SOLAR CNC 165x165-9 Solar panel

3.13.2. Voltage regulation

The powering circuit is composed of an LM7805 fixed linear voltage regulator integrated circuit (Figure 33). This IC has built-in protection against over-current and over-heating issues and are a standard in DC-DC low-cost step-down voltage conversion. The fact that they are linear switching regulators means that the input current and the output current are of the same value even with varying input voltages. This means that there is a heat dissipation associated with the voltage level conversion with power equal to:

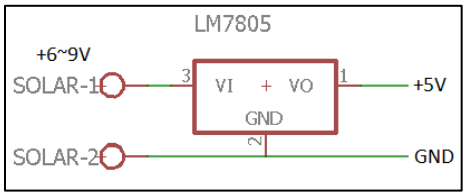


Figure 33 - Voltage Regulation

$$P_w = (V_{in} - V_{out}) * I_{out} \tag{10}$$

Averaging the drawn current to be around 200[mA] and the output solar panel voltage to be 7[V] we get an estimate power dissipation (P_w) of:

$$P_w = (7[V] - 5[V]) * 200[mA] = 400[mW] \tag{11}$$

This heat dissipation can cause overheating which will cause the LM 7805 IC to overheat and shutdown if proper heat dissipation is not ensured. In order to calculate whether a heatsink is necessary, we can perform this simple calculation for the maximum power dissipation at room temperature:

$$P_{wmax} = \frac{T_j - T_{amb}}{R_{\theta JA}} = \frac{175[ºC] - 25[ºC]}{62\left[º\frac{C}{W}\right]} = 2.41[W] \tag{12}$$

In some cases, (such as our own) this means that the efficiency of the power collecting circuit is not as high as it could be with other methods. This part was chosen due to its ready availability and low cost. Since the value for Pw is much smaller than the value P_{wmax}, it is safe to operate the LM7805 without a heatsink.

The produced prototyped circuit can be viewed below [Figure 34].

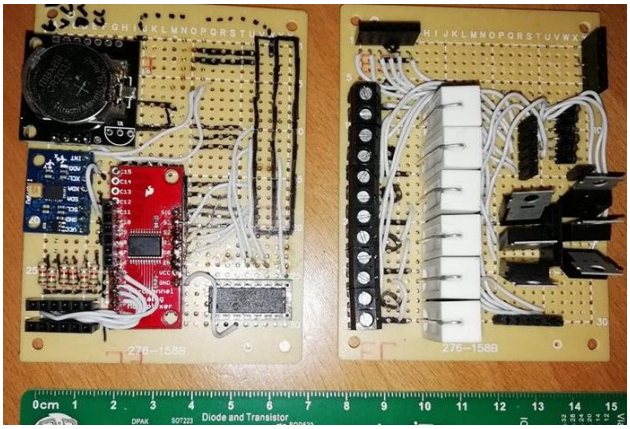


Figure 34 - Final Circuit

3.14. Package installation

The Raspberry pi requires an operative system to function. There are several Linux distributions compatible with the used model of Raspberry Pi. Due to the simplicity of the task at hand, the standard release of the Debian OS was installed in the device.

Due to the specificity of the tasks needed for the proper function of the prototype, some additional functions were installed. These relate mainly to the extra hardware connected to the Raspberry Pi and to tools required for GSM communication.

3.14.1. I2C

To install I2C, the following commands must be inserted in the command prompt:

- sudo apt-get install -y python-smbus
- sudo apt-get install -y i2c-tools

After the installation of these components is complete, we must use the **raspi-config** wizard to enable I2C communication. After this is done, the i2cdetect command can be used for troubleshooting.

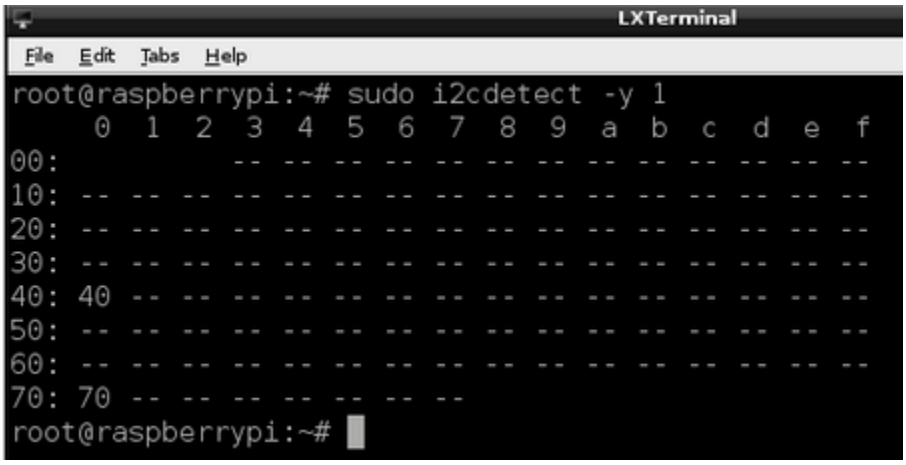


Figure 35 - I2C Address verification

If everything was done correctly, we should see [Figure 35] the entries 50, 68 and 69. These entries correspond to the EEPROM integrated in the RTC, the RTC itself and the IMU, respectively.

3.14.2. SPI

SPI interfacing options are included in the default Debian installation, but there is still the need of activating SPI via the Raspi-config menu [Figure 35].

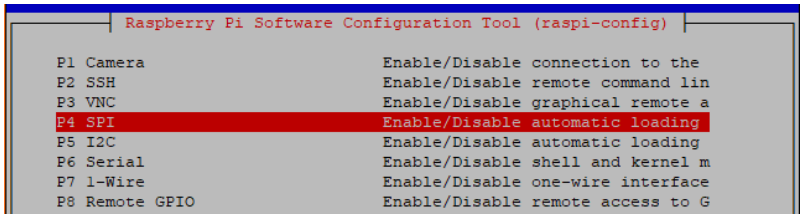


Figure 36 - Raspi-config menu

After this is enabled, we can type **/dev/*spi*** in the command prompt to verify if SPI is active. The system should respond with:

/dev/spidev0.0 /dev/spidev0.1

This means that both SPI channels are up and running.

3.14.3. HWClock

The RTC module must be loaded by the kernel by running: "**sudo modprobe rtc-ds1307"** in the command prompt. After this is done, we can listen to see of the RTC is well connected and ready to function using "**echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device**" in the command prompt.

After this is done, we can check the time on the RTC device using: "**sudo hwclock -r**". If this is the first time the module has been used, it will report back January first, 2000, and we shall need to set the time to the current time. The easiest way is to connect it up to Ethernet or Wi-Fi. It will automatically set the time from the network. You can check the time using the "date" command.

Run "**sudo hwclock -w**" to write the system time to the RTC board. You can then verify it with "**sudo hwclock -r** ":

The next time the raspberry boots, it will not know that it needs to read the clock from the RTC, so if it is not connected to the internet it will be showing the wrong time. It is required to add the RTC kernel module to the file "**/etc/modules"** so it is loaded when the Raspberry Pi boots.

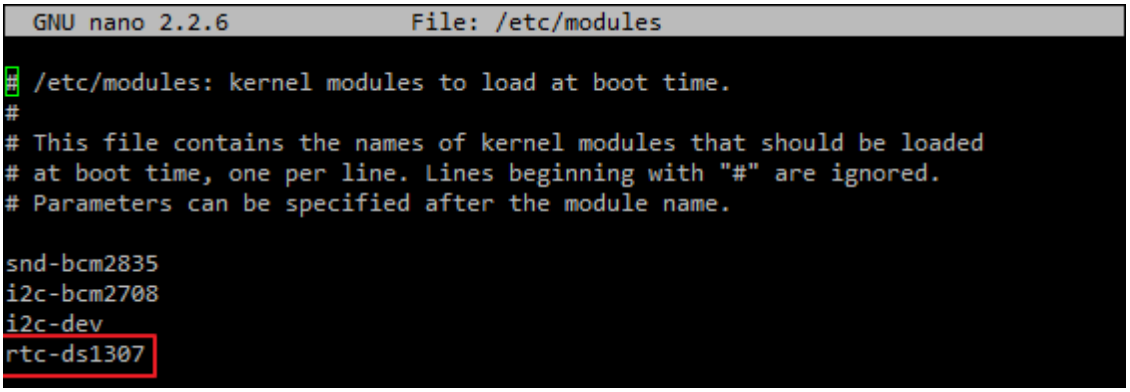


Figure 37 - I2C RTC device registration [1]

Next you will need to add the DS1307 device creation at boot by editing the "**/etc/rc.local**" file by running "**sudo nano /etc/rc.local**" and add the following lines to the file (Figure 36):

echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
sudo hwclock -s

date

The Raspberry Pi is now using the RTC clock as the default time and date and will not require an internet connection to have time and date information. This is very useful to reduce the connection time and to always keep track of time even without communications.

3.14.4. 3G Dongle

The last configuration needed is the 3G dongle. In order to get the Raspberry Pi to connect to the internet via GSM, the first step is to disable the automatic wi-fi network search feature, since it would override the GSM connection. This may be advantageous in some situations since wi-fi consumes less power, but in order to demonstrate the functioning of the prototype in a situation where no wi-fi is available we disabled this feature. In an upcoming version of the project, it may be desirable to implement a conditional connection script that evaluates the different available connections, whatever they may be, and connects to the one which ensures better battery performance.

The USB dongle has two modes: one which installs the GSM software in a computer (provided by the GSM service provider) and one which performs the communications. The software included in the latter mode is not compatible with Raspberry Pi systems, however it is the default operation mode for the device. This must be changed and to do so a tool called **usb-modeswitch** was utilized.

The next step is installing Sakis3G, the software that allows the Raspberry Pi to connect via GSM.

After the GSM is installed, we must run the following command on startup to connect to the internet:

umtskeeper --sakisoperators "USBINTERFACE='0' OTHER='USBMODEM' USBMODEM='12d1:1c05' APN='CUSTOM_APN' CUSTOM_APN='CUBO'SIM_PIN='0000' APN_USER='USER' APN_PASS='PWRD'" --sakisswitches "--sudo --console" --devicename 'Huawei' --log --monthstart 8 --nat 'no'

The command comprises all the information needed to run and start the connection with a predefined custom APM and SIM card, as well as all the information about the device for compatibility. The complete code is annexed.

After all these features are installed, we can run the main.py script which will measure all the sensors and send the information to the Adafruit.IO platform. The main.py code can be found in annex 1.

3.15. Cost analysis.

The required materials for building one instance of the first version of the sensor are listed below (Table 1):

Item	quantity	Price (unit)	Total
Huawei GSM module	1	21,70€	21,70€
Raspberry pi 3 model B	1	39,9 €	39,90 €
Raspberry header	1	2,58 €	2,58 €
Relay module	6	3,69 €	22,14 €
1,5k resistor	18	0,10 €	1,80 €
1,5k thermistor	6	0,10 €	0,60 €
5W 10 shunt resistor	6	0,10 €	0,60 €
LM7805 Voltage regulator	6	0,49 €	2,94 €
4,5W Solar Panel	6	12,10 €	72,60 €
screw clamp	6	0,86 €	5,16 €
16 channel analog multiplexer	1	5,00 €	5,00 €
Real time clock	1	5,90 €	5,90 €
MPU 6050 breakout	1	8,36	8,36 €
White wire	1	3,00	3,00 €
Solder wire	1	5,60	5,60 €
mcp3008 ADC	1	4,50	4,50 €
20800 mAh Power bank	1	45,00	45,00 €
Junction	12	0,35	4,20 €
Custom PVC casing	1	30,00	30,00 €
Total	-	-	271,58 €

Table 1 - Prototype costs [11-10-2018]

The cost of this first prototype was of 271,58[€]. This value has the potential to be largely reduced with design improvements, different configurations and scaling up production numbers, this matter is addressed in the final chapter of this work.

4. Experimental Results

4.1. Power Consumption analysis

Upon completion of the circuit and programming, the power consumption of the prototype was analysed. The objective of this procedure was to determine the ideal size of the battery pack. To assess the average current draw of the device, several measurements were made: one when the device is waiting for the next measurement and one when the device is communicating. For practicality, some assumptions were made. These assumptions are as follows:

- The device spends on average 3 seconds of each minute communicating and the remainder 57 seconds waiting for the next measurement.
- The startup current draw from when the device is booted up is ignored.
- The current and voltage at the device's terminals are constant during the measurements.
- The device is not running on battery power.
- The measurement of the different inputs does not have an effect on power consumption.
- The device runs on 10V DC power supplied in parallel to the six solar panel inputs.

For a good benchmark, several features of the Raspberry Pi were turned off progressively, in order to assert which of the features have higher demand. The test setup is as follows (Figure 37):

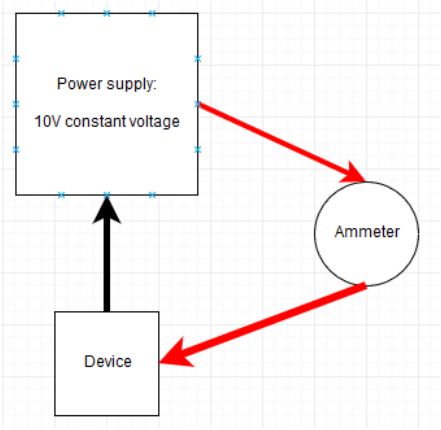


Figure 38 - Consumption testing

The results from this process were as follows:

Test	Current [mA] (Standby)	Current [mA] (Communication)
Raspberry Pi	430	600
Raspberry Pi (No HDMI)	380	600
Raspberry Pi (No Wi-Fi)	330	600
Raspberry Pi (No sound)	320	600

Table 2 - Current draw test results

All the results above [Table 2] are average consumption values, since these values are prone to very fast fluctuations that cannot be measured without specialized equipment. Upon measuring the consumption of the device, it becomes clear that too much power is being consumed. For the lowest estimate, the required capacity for a 3 days operation on battery power is given by equation 13.

$$C[mAh] = 3[days] * 24[h] * \left(\left(\frac{3}{60}\right) * P_{com}[mA] + \left(\frac{57}{60}\right) * P_{standby}[mA]\right) \quad (13)$$

The required capacities for each of the different configurations are [Table 3]:

Test	Capacity[mAh]
Raspberry Pi	31572
Raspberry Pi (No HDMI)	28152
Raspberry Pi (No Wi-Fi)	24732
Raspberry Pi (No sound)	24048

Table 3 - Capacity calculations

The battery provided for this project was of 20000[mAh], which means that even for the best estimate, the autonomy of the device would be less than 3 days.

4.2. Cube placement and location

The cube was placed on the roof of building C1 in the Faculty of Sciences of the University of Lisbon (indicated by the red circle in figure 38) with its front face towards south. The precise location of the cube is 38°45'22.8"N 9°09'24.4"W [Figure 39]. The first test was conducted for four days, starting 31st of August, 2018 and ending in 3rd September, 2018. During these four days, two regimes of meteorological conditions were observed, three clear days and one cloudy day. This is important to determine the behavior of the sensor and the precision of the measurements with different climate conditions.

Figure 39 - Cube location [14]

Figure 40 - Final Product

4.3. Raw Short-circuit current data analysis.

The first piece of meaningful information comes from observing the raw short-circuit current data from the different faces of the cube. As expected, all the faces the short circuit current is directly proportional to the amount of irradiation on each face. The left face reaches peak irradiation values in the morning period (around 2 hours after sunrise). The right face has the opposite behavior, reaching its peak value around two hours before sunset. (Note: During some periods of the morning and near the sunset there was a shadowing of the device resulting in abrupt drops in the measured values. Better device positioning would prevent this from happening.) The top face is exposed to the same irradiation as a horizontal pyranometer, and the front face is exposed to the irradiation of a south facing vertical pyranometer. The bottom face measures no irradiation since the sensor was placed directly on the roof of the building (Note: Applying a solution where a face faces towards a surface can generate data on the albedo of that surface, hence the bottom facing face of the cube could be used to measure roof albedo). Lastly, we have the back-facing surface which has the same irradiation values of a north facing vertical pyranometer. (Figure 39)

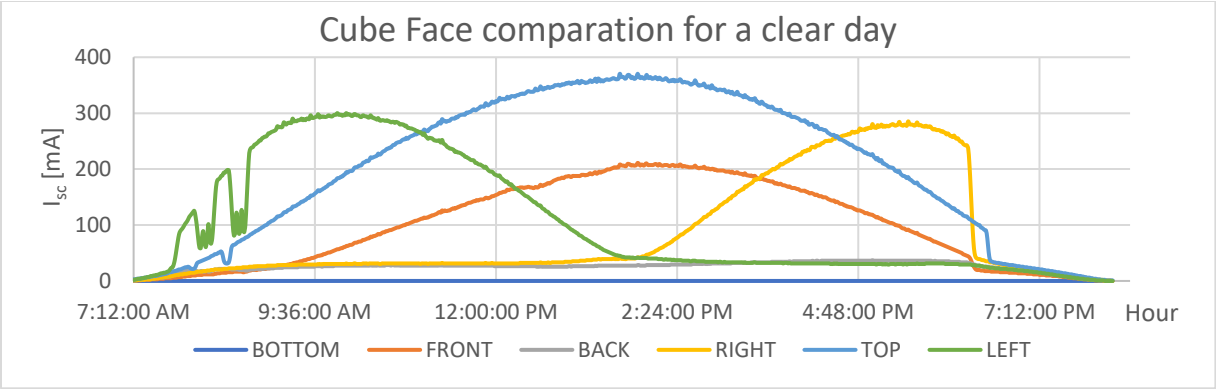


Figure 41 - I_{sc} for the different faces of the cube

Using the times at which the maximum of each face is reached, we can calculate the cube's orientation. In the case of the cloudy day, conclusions cannot be taken so lightly due to the variability of cloud cover (figure 38). There are however some interesting phenomena. Looking closer at the peaks in some faces we can see that there are instants where the irradiance on some faces is greater than the value of irradiance for the same face of the cube for a clear day, this means that the cube may be exposed to irradiation cloud enhancement events [Figure 40].

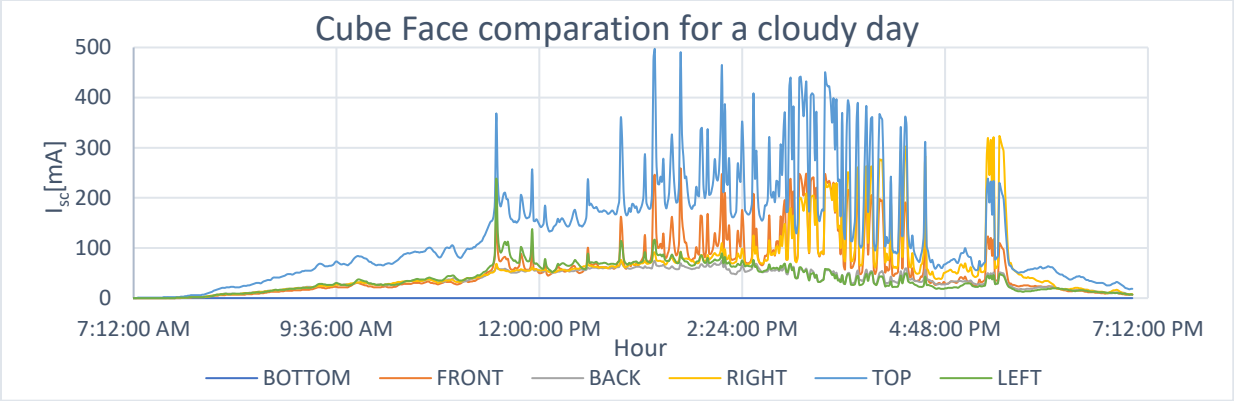


Figure 42 - Cube Face comparation for a cloudy day

4.4. Comparison between the Cube and the meteorological station

Next to the sensor cube, there is a solar resource evaluation installation comprised of a horizontal pyranometer and a shaded horizontal pyranometer, which gathers data relative to diffuse and global irradiance (MS-802 pyranometer). The first data to analyze is whether I_{sc} data form the cube's solar panels can be used to reliably infer GHI measurements in the meteorological station. For this, only the data from the top face of the cube and from the station's horizontal pyranometer were used. For a clear day, the resulting data is as follows (Figure 41):

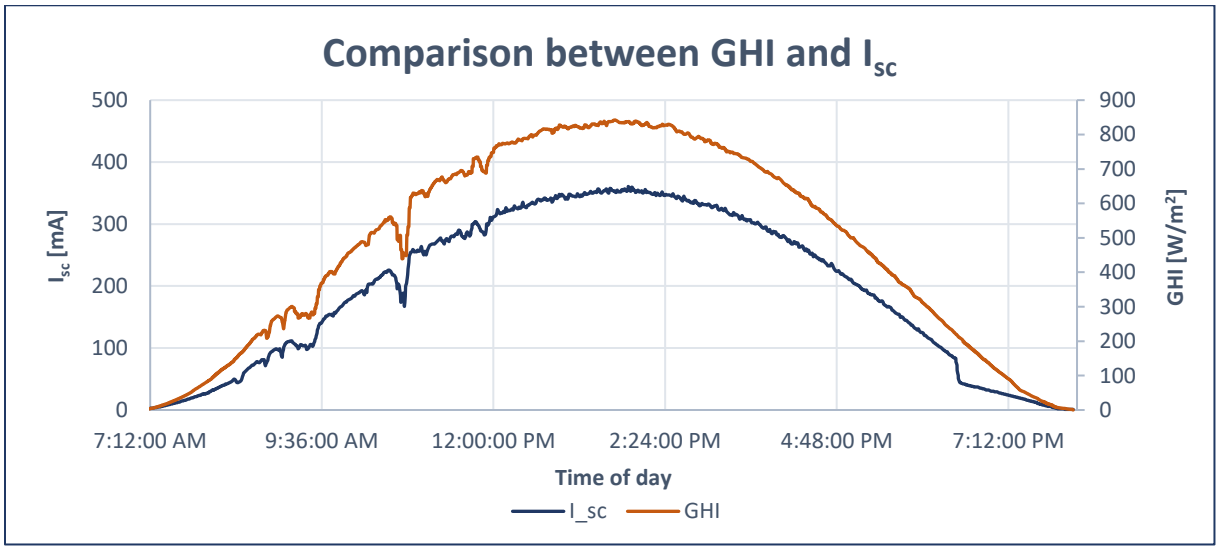


Figure 43 - Comparison between GHI and I_{sc} for a clear day

As confirmed by figure 42, the two sensors are very similar and have a correlation of 0.9933. This means the data from our prototype has an accuracy comparable to the accuracy of the meteorological station.

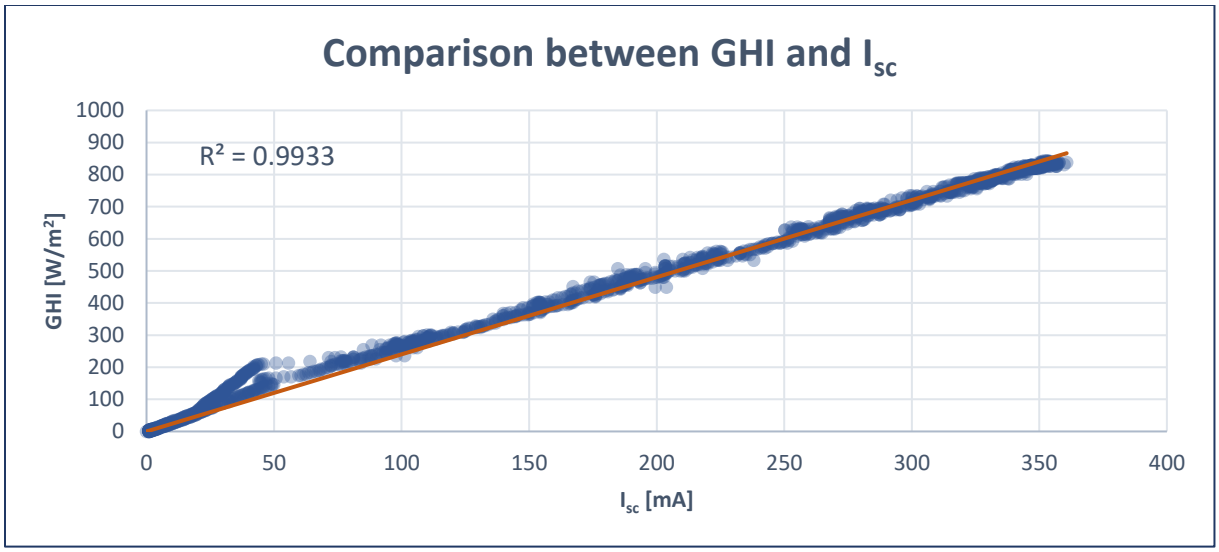


Figure 44 - Correlation between measured I_{sc} and GHI for a clear day

The same comparisons were made for the cloudy day (figures 43 and 44).

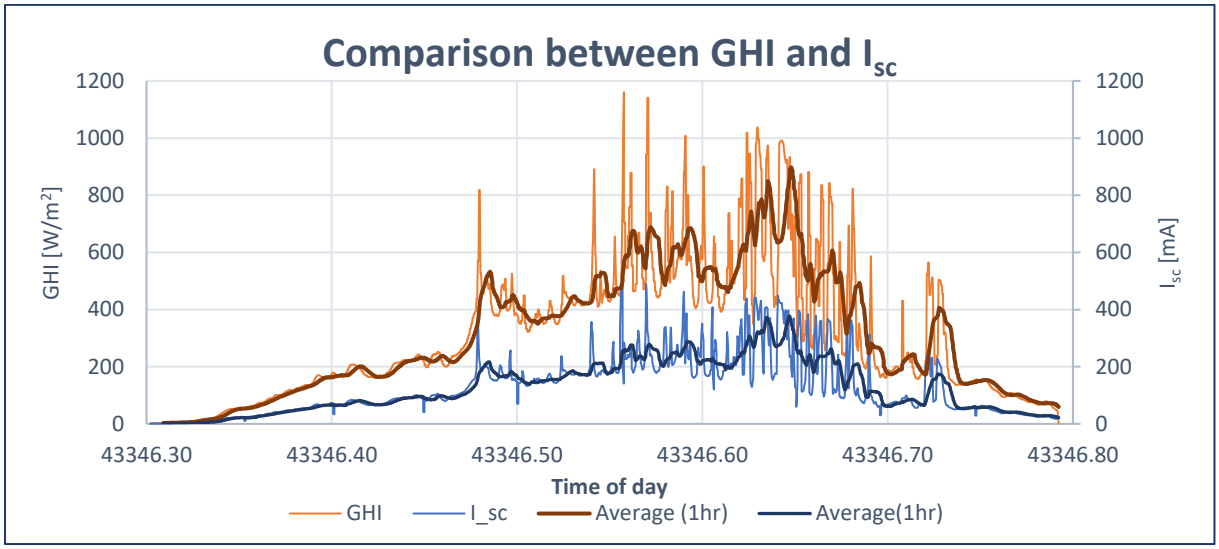


Figure 45 - Comparison between GHI and I_{sc} for a cloudy day

As confirmed by figure 41, the two sensors are very similar and have a high correlation of 0.9615 even for cloudy days, allowing us to further the latter conclusion. Differences are attributed to different angular responses (solar modules are more sensitive to the angle of incidence) and spectral responses. As mentioned in section 1.1, if the application is the assessment of PV potential or forecast, it might be useful to measure irradiation using a solar module rather than a standard pyranometer.

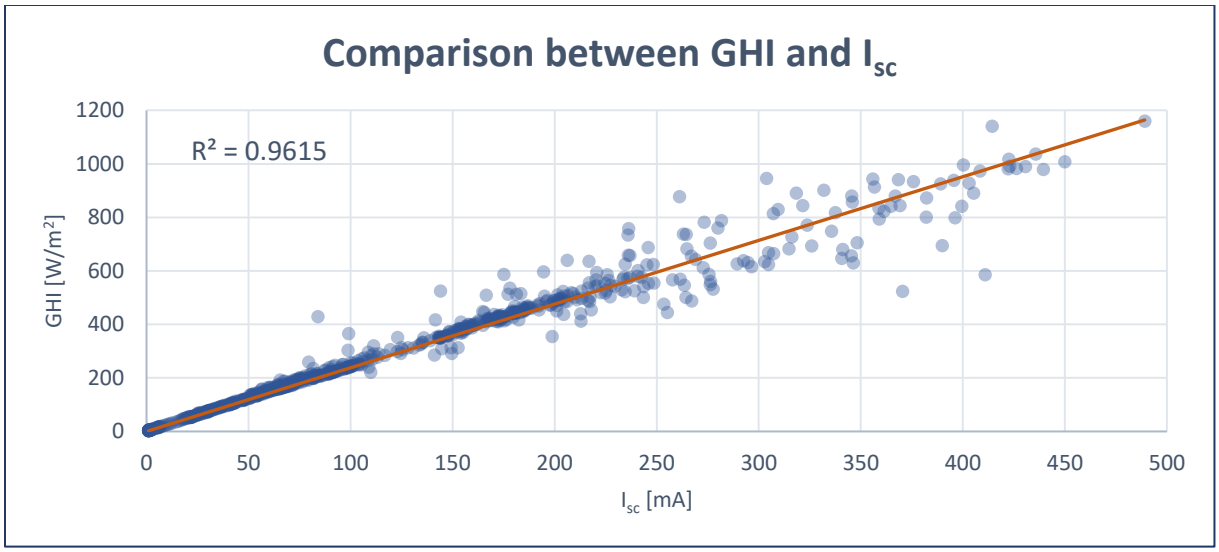


Figure 46 - Correlation between measured I_{sc} and GHI for a cloudy day

This concludes the first approach to the data produced by the cube. This data allows us to say with a relatively high certainty that the I_{sc} in the top facing solar panel is a good substitute for the pyranometer. This is an important step because the prices of the two components are very different and despite, they yield the same results independently.

5. Artificial neural network usage to estimate solar radiation components

5.1. Spatiotemporal solar forecast

With the rise in the percentage of energy coming from solar power, knowing how much power is being produced and where, becomes of paramount importance to ensure the power grid is stable. Installing production monitoring hardware and software solutions represents an added cost to PV. It is simpler and cheaper to calculate the production of a set area using forecast models. These models must consider the spatial and temporal variability of solar power production. Systems like Sky-cam [Figure 45] and other similar systems use these methods to infer information relative to the size, position and speed of clouds travelling through the sky, so that they can then extrapolate the power output of solar power installations in the vicinity.

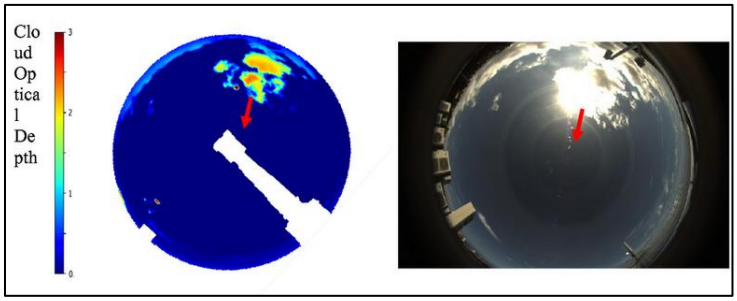


Figure 47 - Example of Sky-cam usage [15]

5.1.1. Machine learning

Machine learning (ML) is a field of computer science that uses statistical techniques to give algorithms the ability to progressively improve performance on a specific task. ML explores the construction of algorithms that can make predictions on data. The ML technique used in this project is named Artificial Neural Network (ANN). It is an array of elemental building blocks called perceptrons, which are algorithms built to emulate the function a neuron accomplishes in organic life. The neuron is composed of an axon, which carries information to a neighboring neuron or effector from the neuron itself, a center which processes information, and dendrites which carry information into the neuron from the last neuron or sensory cell. In semblance to the function of the neuron, the perceptron has two connections, one inbound and one outbound, and one activation function which performs an operation on the received data before transmitting it to the next perceptron.

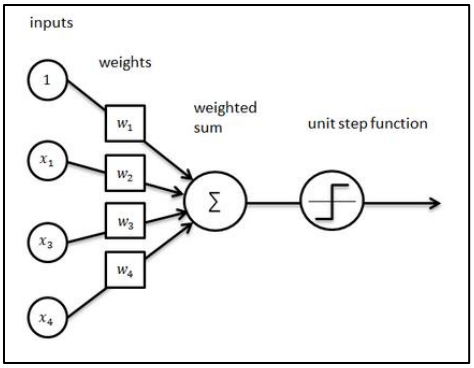


Figure 48 - Example of a perceptron [16]

In order to learn, a biological brain must strengthen the most used connections and let the unused neurons wither away. The rate at which this process occurs is called cerebral plasticity and is related to the speed at which information is learned and forgotten in a biological brain. As the artificial learning process unfolds, the different weighs and biases are changed in order to maintain only the biases which are required for the perceptron to perform its given task. There are two values that change within a perceptron: the weights and the biases. Each connection between two perceptrons has a weight associated with it. And each perceptron has a bias associated with it. In order for the neural network to learn, a cost function is implemented, in similarity with the human brain whose cost function is the energy requirements of having neurons functioning. The cost function is a n-dimensional error function (where n is the product of the number of connections and the number of perceptrons). When the minimum value of this function is achieved, learning stops, and we can use the resulting mathematical model to make predictions.

5.2. Irradiation data

The objective of our project is to estimate Direct Normal Irradiation (DNI) and Diffuse Horizontal Radiation (DHI) from Global Irradiation measurements on different surfaces. Since the prototype was not developed at the earlier phase of the project development, when this approach was being tested, instead of using measured data from the prototype, data from the National Renewable Energies Laboratory (NREL) from the year 2014, with a sample interval of one minute was used.

The NREL data is collected atop a mesa with coordinates: Latitude: 39.742o North; Longitude: 105.18o West; Elevation: 1828.8 meters AMSL. The data is available from dates ranging from July 15, 1981 to the present. The sensors present on the station are: LI-200 for GHI measurement, Li-201 for DNI measurement and SPN1 or CM22 for diffuse radiation measurement. The SPN sensors are oriented in accordance to the four cardinal points, providing a unique dataset for comparing with the solar cube prototype.

5.3. Data pre-processing

For the correct operation of our ANN, the data must first be pre-processed to facilitate the calculations in the learning process. This ensures that there is a higher chance that the network learning algorithms will converge to a global solution faster and that the network is only learning the situations where predictions will be made.

For this, we excluded all data referring to night, dawn and dusk periods. To differentiate these data points, we postulated that every time where the zenith angle is above 80° is considered belonging to the night,

dawn and dusk period and is thus eliminated. This was done because these times of the day are where there is more risk of shadow interference and the forecast at this time of the day is unnecessary.

Upon analysis of the imported data we also found some time periods where measurements surpass the value of 1500 [W/m2]. This is extremely unlikely, usually due to measurement error; therefore, these data points are also discarded. Equally unlikely are data points where the diffuse normal irradiance (DNI) and the direct horizontal irradiance (DHI) or their sum are larger than the value of the global horizontal irradiance. These values are also discarded. There is however a chance of having global irradiance levels higher than the solar constant, due to cloud reflection phenomena concentrating light on the sensor. This is relevant data and is not ignored.

The final step is to separate the data into three sets: training, testing and validation. The training set is used to make a random neural network. This network is then asked to make predictions with the test sets. These predictions are then compared to the validation set to get an error estimation. The objective of the neural network is to adapt its weighs and biases in order to minimize the error. When the error is below a certain value, when it ceases variation or when a set number of calculations is reached, the converging process stops. The resulting weighs and biases comprise the neural network model to be used in the forecast process.

The typical way of going about this final step is to divide the data randomly as to avoid any bias caused by the order in which the data is presented to the network. This means that 70% of the data goes into training, 15% into testing and 15% into validation. But this still means that there is a chance that our network will be testing summer data against winter data and so on. To counteract this, the data is divided into chunks of 9 consecutive blocks of data (9 minutes). For each of these chunks we define that the first 5 belong in the training set, and the other 4 data are divided for test and validation (2 data for each), each having a different index variable. This ensures that our neural network is going to factor in the contributions of the seasons and prolonged weather patterns in its estimates.

	OUTPUT		**INPUT**			
Test	DNI	DHI	GHI[W/m^2]	Azimuth[º]	Zenith[º]	GHI[W/m^2] facing N,S,E,W
A	X		X			
B	X		X	X		
C	X		X	X	X	
D	X		X	X	X	X
E		X	X			
F		X	X	X		
G		X	X	X	X	
H		X	X	X	X	X

Table 4 - ANN Test configurations

To determine the correct size of the neural network, two different numbers of neurons were tested. For each of the network sizes (5 and 15 neurons), the inputs and outputs were varied along 16 different tests labelled from A to H as seen in Table 4. After the ANN makes its estimates, we apply a filter where any negative value [W/m2] is discarded as a faulty estimation. The output of the ANN script is a set of mean square error [RMSE] values for each of the components, train, test, validation and overall error.

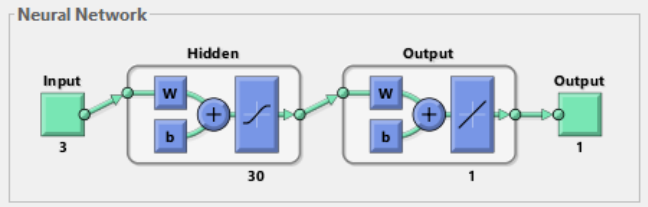


Figure 49 - Neural Network Representation

Comparing the RMSE values for the tests with 5 hidden neurons and 15 hidden neurons, we can see that for the simplest tests (A and E) there is no difference in using 5 or 15 hidden neurons. What this means is that for the simplest tests there is no need for more than 5 hidden neurons for the model to comprehend the situations presented to it due to the low number of inputs. On the opposite side, the RMSE values for the most complex tests (D and H) show some differences, meaning that this model is now too complex to be comprehended by 5 neurons. This also means that further testing must be done with more neurons before conclusions may be had. [Figures 48 and 49]

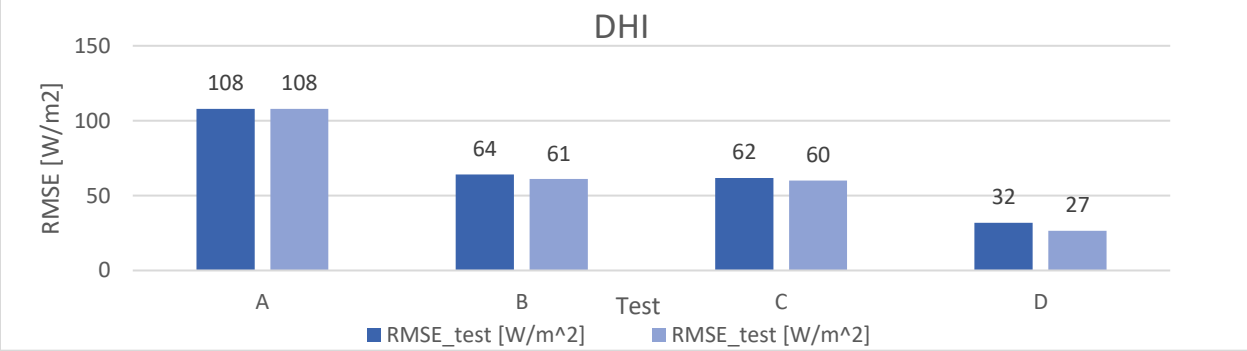


Figure 50 - RMSE comparison between 5 neuron models and 15 neuron models for DHI

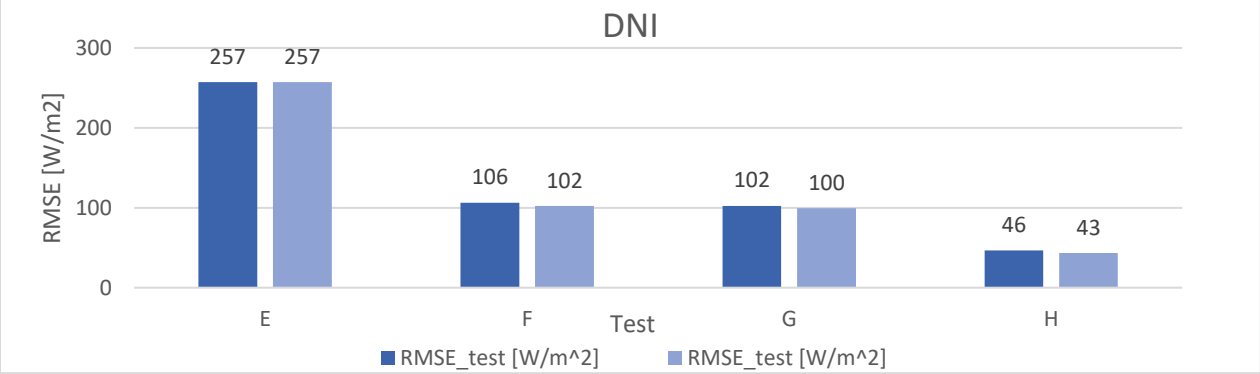


Figure 51 - RMSE comparison between 5 neuron models and 15 neuron models for DNI

The main purpose of the first set of tests was to determine the importance of the number of neurons and the correct input data to train the model. In the next section the results of the different ANN training processes will be compared for different sky conditions.

5.4. Results

For a clear day, the observed GHI follows a constant pattern. This pattern is correlated with the sun's position in the sky, varying only with the hour of the day and the day of the year. This can be observed in

the leftmost chart [figure 50]. For a day with a light cloud cover, the overall shape of the GHI curve generated by the sun's movement across the sky can still be observed, but with subtracted values originating from the clouds obstructing the sun disk. In this situation, the irradiance comes from both DNI and DHI components. This situation can be observed in the centre chart [Figure 50]. For a day with a heavy cloud cover, the pattern is completely irregular and its evolution throughout the day is not related to the sun's movement across the sky but rather from diffuse light coming from different parts of the cloud cover. The irradiance in this case comes only from the diffuse component. This can be observed in the rightmost chart [Figure 50].

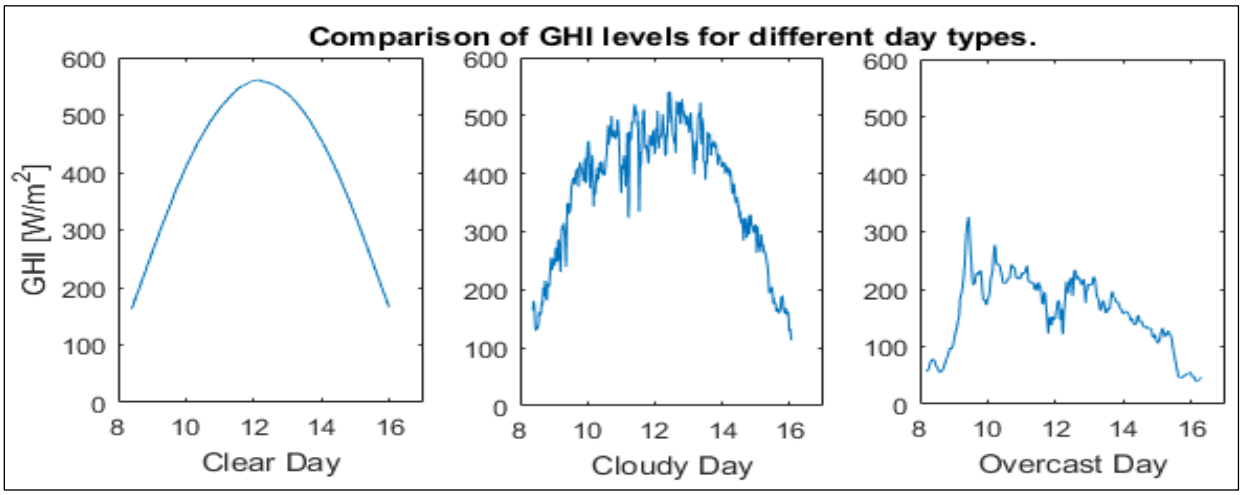


Figure 52 - Comparison of GHI levels for different day types

Using information on the results of the ANN, it is possible to make predictions to estimate the direct and diffuse components of sunlight using global irradiance values. Some days of the validation data from the ANN were chosen to illustrate how the different ANN models developed in the latter portion of this work react when trying to predict direct and diffuse irradiance components for days with different types of cloud covers.

Analyzing [figure 44] we can conclude that on days with clear sky conditions, the levels of DHI are much lower than in cloudy sky conditions, increasing the relative error between the estimated and measured

data. Here we can see that the estimates using all available data (Test D) are significantly better than estimates using only GHI, Az, Z and KT data (Test B).

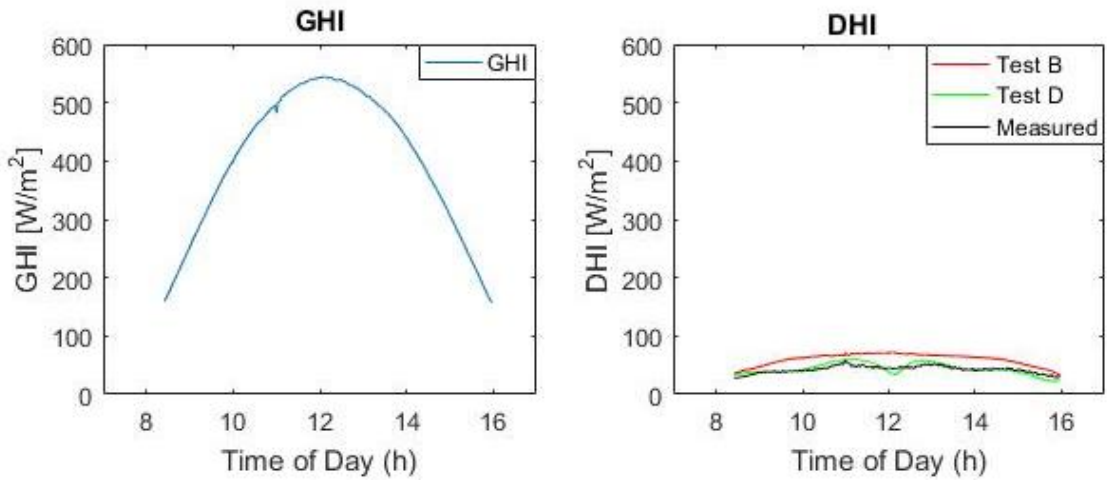


Figure 53 - DHI estimation comparison with models F and H for a clear day (March 12th 2014)

Looking at DNI values for the same day [Figure 51], we can see that the ANN with more inputs (Test H) is the one with the best forecasting behavior, we can also see that for clear days the ANN output is overcomplicated, and the ANN fails to comprehend the simple case of a clear day. This problem can be overcome by utilizing a classification algorithm to choose between clear and cloudy days and training a different ANN for each subset.

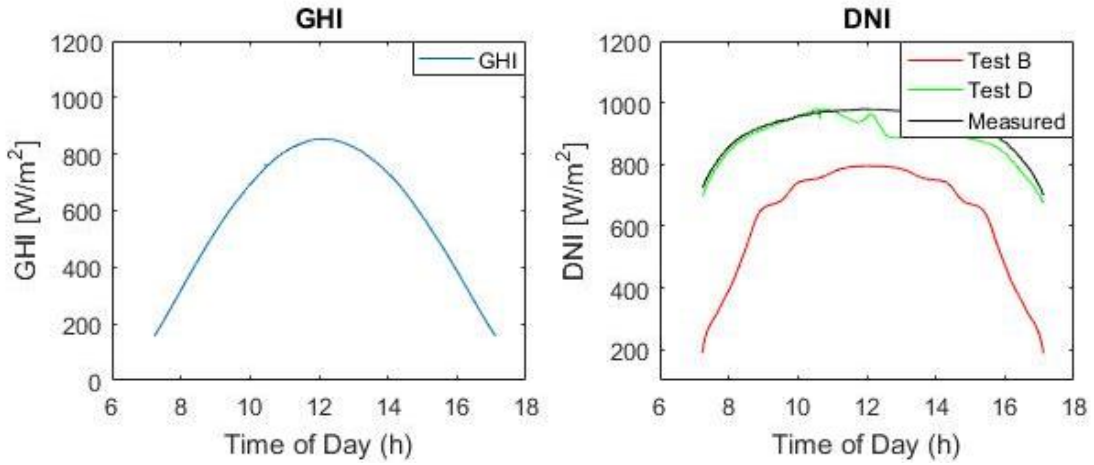


Figure 54 - DNI estimation comparison with models B and D for a clear day (March 12th 2014)

Analyzing [figure 52] we can conclude that on days with cloudy sky conditions, the levels of GHI are almost entirely due to diffuse irradiance (DNI), this is confirmed by [figure 53] where the absence of direct

irradiance can be observed. This resemblance means that if the ANN has enough information to determine whether the day being analysed is a cloudy day or not, it will reliably determine the value of DHI from GHI.

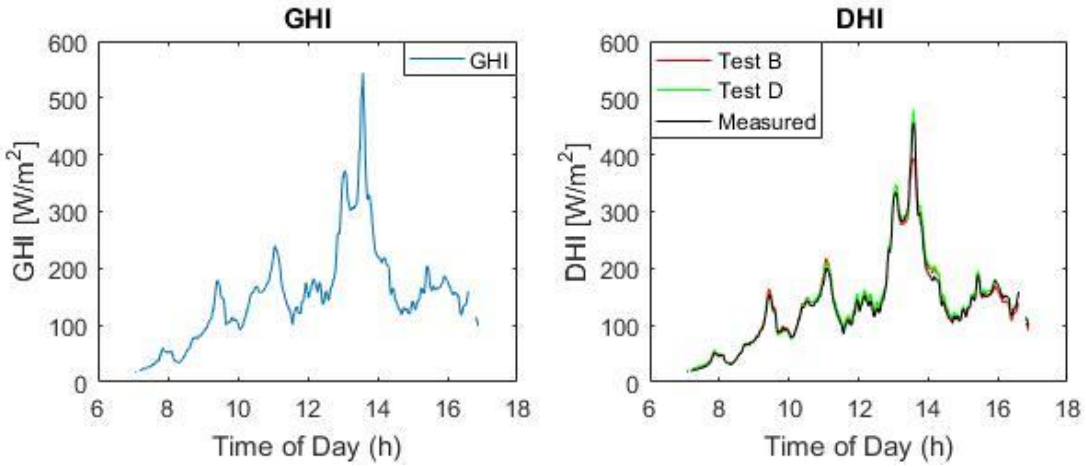


Figure 55 - DHI estimation comparison with models F and H for a cloudy day (April 27th 2014)

Looking at DNI values for the same day [Figure 54], we can see that the ANN with less inputs (Test H) struggles in predicting a scenario where there's almost no DNI. The model with more information accurately predicts this scenario. This problem can be overcome by utilizing a classification algorithm to choose between clear and cloudy days and training a different ANN for each subset.

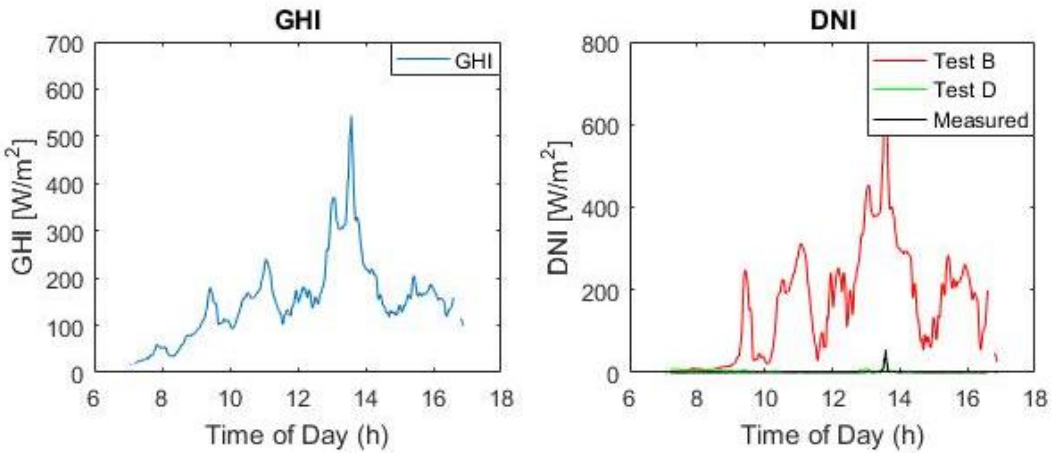


Figure 56 - DNI estimation comparison with models B and D for a cloudy day (April 27th 2014)

The resulting ANN model (Test D) can predict both components of the global irradiance with high accuracy. Figures 55 and 56 represent the comparison between measured and real irradiance values. The high values of R^2 indicate a strong correlation between the sensors and the ANN output, meaning that the ANN can become a good substitute for the direct measurement of all irradiation components.

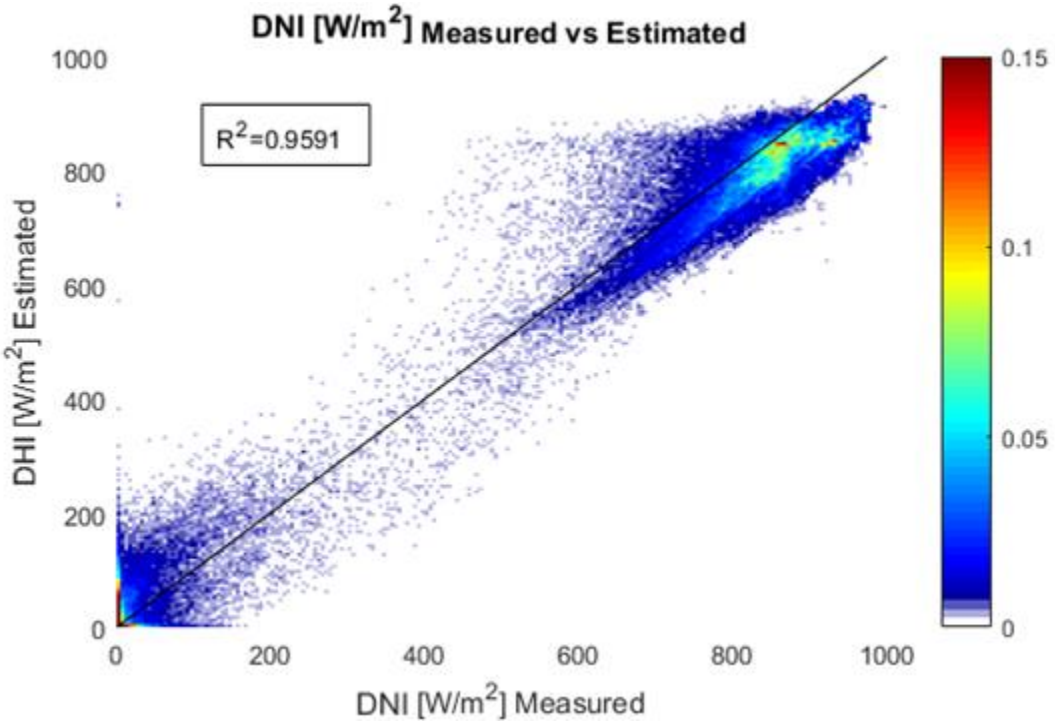


Figure 57 - Density plot showcasing the estimated DHI from GHI, Azimuth and Zenith data

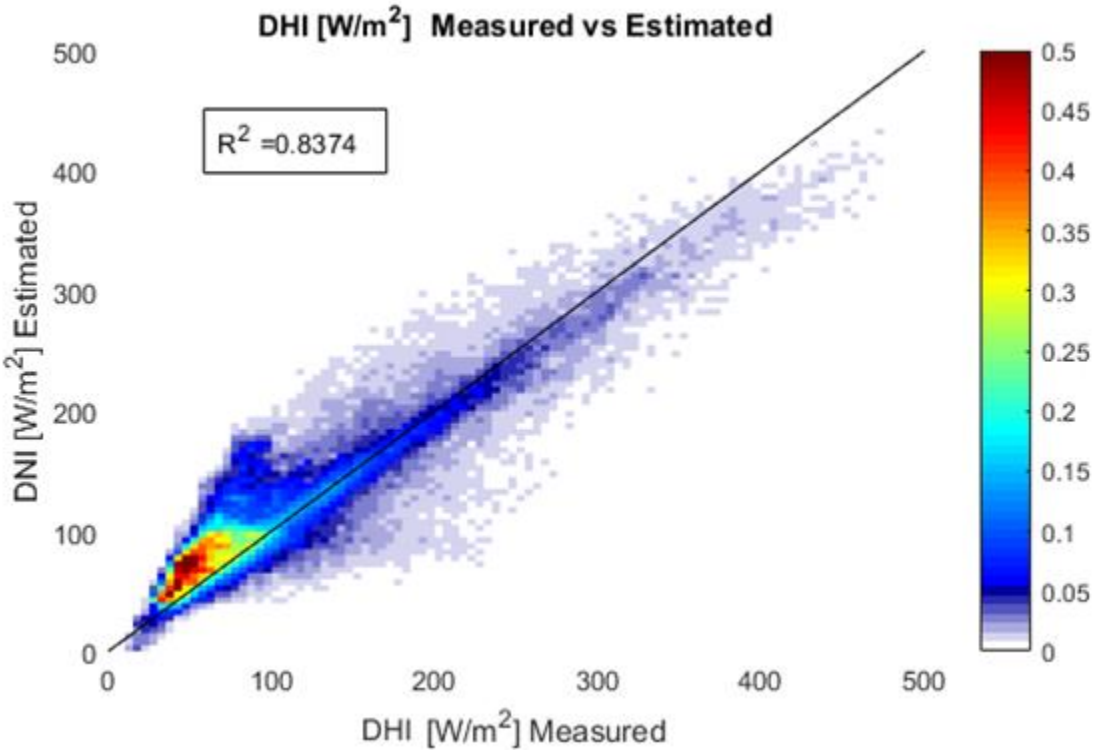


Figure 58 – Density plot showcasing the estimated DNI from GHI, Azimuth and Zenith data

These results are a strong indicator that there is a strong (yet a priori undetermined) correlation between the global irradiance seen on surfaces with different orientations and the diffuse and direct horizontal irradiance. Furthermore, this technique may prove useful for detecting clouds due to shading and reflection patterns on the cube's faces, although this was not tested, since it requires spatially disperse sensors that were not available. Adding a second ANN which is fed the data from the spatially dispersed sensors can be an accurate means of predicting cloud travel paths and consequentially improve the sensitivity of solar production forecast systems.

From the operational point of view, it is important to mention that this approach requires a data set for training the ANN. In a scenario of large-scale deployment of Solar Cubes in a town, it would still be required to measure irradiation using standard methods at one, or more, sites, at least to provide the data for the training step. This data collection could be done for a limited period or, preferably, continuously, for the regular performance monitoring and eventual upgrade of the ANN model. If taken in different sites, representative of the region, comparison between sites would validate the assumption of using one single ANN model for all Solar Cubes deployed in the region.

6. Discussion and Outlook

6.1.1. Arduino based microcontrollers.

For the initial purposes, there were four potential choices for the microcontroller within the Arduino catalog. All these MCUs are part of the same line of products based on the Arduino MKR Zero board (Figure 13) and vary only in the communications system. The Arduino MKRZERO is a low cost all-round MCU capable of both digital and analog measurements as well as standard GPIO, I2C, Serial and SPI interfaces. Based on the SAMD21 Cortex-M0+ 32bit low power ARM MCU processor, it comes equipped with a built-in charge controller for li-ion batteries and an internal RTC, making it ideal for self-powered off-grid operation. The other potential MCU modules are listed below and differ from the MKRZERO only in their wireless connectivity features:

- Arduino MKR 1000 (with integrated Wi-Fi module)
- Arduino MKRFOX 1200 (with integrated SIGFOX module)
- Arduino MKRWAN 1300 (with integrated LoRa module)
- Arduino MKRGSM 1400 (with integrated GSM module)

From the list above, there are two entries which are easily dismissed. Although the Arduino MKR 1000 module uses the widely-spread Wi-fi technology, it does not meet the requirements for long range communication (typical it is less than 100 meters), thus making it unfit for this project. The high-power demand (2.5 W) in GSM communication is the reason why the Arduino MKRGSM 1400 option is being discarded. This decision is also to prevent the cost of mobile communication tariffs on sending data via GSM. The two remainders are very similar in terms of power consumption (20 mA for standby operation and 750 mA peaks for communication) and application, differing mainly in coverage, data tariffs, data transmission speed and communication protocols. We still need to decide between LoRa and SigFox.

Figure 59- Arduino MKRZERO [17]

6.1.1.1. LoRa

Initially the two remaining MCUs were acquired for further testing and to start the software development process. The LoRa module has limited coverage, that stems from the way LoRa networks are created and how they grow in size. LoRa users must either build their own antennae (called gateways) in order to communicate with the internet or purchase the service from a provider. In Lisbon, there are currently a limited number of existing networks and most of them belong to individuals or closed research groups, rendering this option inapplicable for the city of Lisbon.

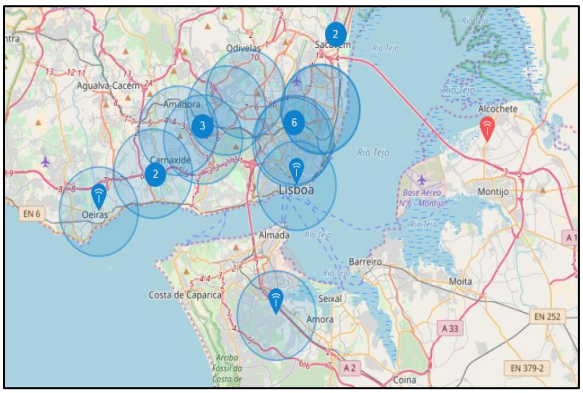


Figure 60 -LoRa coverage in Lisbon [18]

6.1.1.2. SigFox

SigFox on the other hand, requires no extra hardware to function and offers a much wider coverage, but the network is private and requires the purchase of a communications package from NarrowNet (the service provider in Portugal) which will put its costs near the cost of GSM. Aggravating this, is the fact that using SigFox, the user is restricted to six communications per hour, which is a lower sample rate for the purposes of this project, thus rendering the Arduino MKRFOX 1200 unfit for usage in this project.

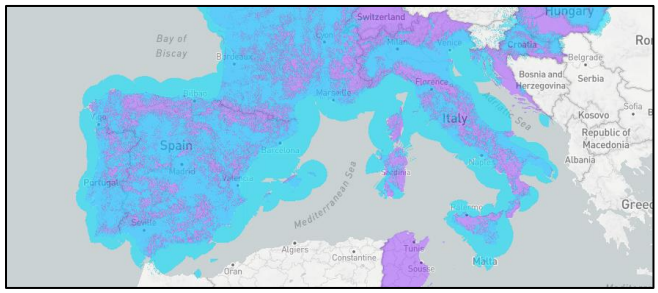


Figure 61 - SigFox coverage in southwestern Europe [19]

6.1.1.3. GSM

GSM communication has been a standard in long range standalone IoT projects. Although it features a higher energy consumption, the readily available and relatively cheap service makes this the best option for now, while other technologies with less power consumption are still in developing and deployment stages. Therefore, the Arduino MKR GSM 1400 is the selected microcontroller for the next phase of the prototype.

6.2. Power management

Opting for an Arduino based system allows us to reduce the idle consumption drastically. Systems like the one used in the prototype always require an operative system to be running constantly. Opting for a simpler processor decreases the energy demand while idle, allowing us to reduce battery size. Linear Voltage regulators like those used in this prototype have very limited efficiencies that depend on the input and output voltages. As a rule of thumb, the efficiency of a linear regulator can be estimated by the ratio of the input and output voltages. This means that in our case, 50 % of the input power is being dissipated as heat before it ever reaches the rest of the circuit. A buck converter like the LM2596 (Figure 60) circuit would be a good alternative to this since it can perform the same regulation (from 10V to 5V) with efficiencies of up to 85%. Thus, using a Buck converter in our setup would almost halve the capacity required for 3 days of battery autonomy.

The Arduino MKR GSM 1400 has a base consumption of 20[mA], with bursts of 700[mA] when communicating. This fact alone makes our predicted capacity for 3 days operation (using equation 13) to be about 10 times smaller than the worse estimate for the Raspberry Pi at 3528 [mA].

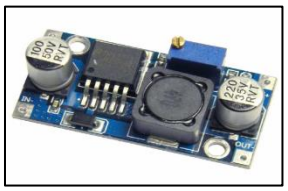


Figure 62 - Step down buck converter [20]

Furthermore, a solar charge controller like the MCP73871 solar charger with MPP tracking capabilities, enables us to preforme pass through charging, which was impossible using the initial powerbank, since its internal safety features does not allow simultaneous consumption and charging.

The TPL 5110 timer allows us to further decrease the power consumption of the device. The device is in idle state for 95 percent of the time. With this type of approach it is possible to have the device only turn on when measuring and communicating, bringing the required capacity for 3 days of autonomy to 2160[mAh], representing a 30% improvement. These improvements mean that the size and number of the solar panels can also be drastically reduced. Finally, adding a better IMU with an internal magnetometer allows the device to verify its orientation. This is useful in cases where the final positioning of the sensor is uncertain or if the sensor is applied to a moving object, such as a vehicle or a solar tracking platform.

6.3. Different Solar Panel Combinations

Upon analysing the output of the device, it is clear that depending on the final purpose of the project, different solar pannel placements can be more cost effective. Assuming that the information regarding the direct and diffuse irradiance can be inferred using only three solar panels (Left, Right and Top), since there is no irradiance on the bottom and back pannels, and that the front pannel has virtually the same behavior as the Top panel. This implies a direct reduction of the projects cost. Furthermore, applications where only the global irradiance is to be measured can benefit from having only the Top pannel, meaning even more competitive costs. The prices shown here are for a single purchase and do not reflect the price reduction due to high volume purchases nor the price of producing the printed circuit board and the weatherproof casing, although with production of high volumes, this price is expected to drop drastically (about 30% to 40%).

item	quantity	price	Total	quantity	price	Total	quantity	price	Total
Arduino MKR GSM 1400	1	35,00	35,00	1	35,00	35,00	1	35,00	35,00
Arduino headers	1	2,58	2,58	1	2,58	2,58	1	2,58	2,58
solid state relay	6	1,95	11,70	3	1,95	5,85	1	1,95	1,95
1,5k resistor	18	0,10	1,80	9	0,10	0,90	3	0,10	0,30
1,5k thermistor	6	0,10	0,60	3	0,10	0,30	1	0,10	0,10
5w test resistor	6	0,10	0,60	3	0,10	0,30	1	0,10	0,10
Charge controller	1	17,40	17,40	1	17,40	17,40	1	17,40	17,40
solar panel	6	12,10	72,60	3	12,10	36,30	1	12,10	12,10
screw clamp	6	0,86	5,16	3	0,86	2,58	1	0,86	0,86
Analog Multiplexer	1	5,00	5,00	0	5,00	0,00	0	5,00	0,00
TPL timer	1	4,95	4,95	1	4,95	4,95	1	4,95	4,95
MPU 9250 breakout	1	15,74	15,74	1	15,74	15,74	1	15,74	15,74
1500mAh	1	9,95	9,95	1	9,95	9,95	1	9,95	9,95
Total	-	-	183,08	-	-	131,85	-	-	101,03

Table 5 - Future Versions costs

7. Conclusions

The scope of this project was to assess the development a low-cost multi-direction irradiance sensor, performing a detailed cost and performance analysis as well as a comparation between this sensor and the standard sensors used nowadays. The sensor should be small, self powering through PV and provide measured data via long range wireless communication with a sample rate of 1 to 5 minutes. The device should be able to gather the sufficient data in order for a solar resource analysis to be performed.

The sensor is configured as a cube with one solar panel in each face. This configuration allows us to collect data on irradiance form all directions. The device encompasses six 4.5[W] solar panels, equiped with analog thermometers, analog short circuit current readings across all panels aswell as tilt sensing, timekeeping features and a Raspberry pi equipped with a GSM module for processing and communication. The device is powered by a 20800 [mAh] powerbank module that gets charged via the solar panels. The device alternates automatically between charging and measuring states at set time intervals to provide as much battery life as possible.

The sensor was tested during sunny and cloudy days in order to collect a vast range of possible cloud conditions. Information collected in these tests was then compared to the output of the University of Lisbon's solar resource monitorization array for validation. According to this comparison, there is a very high correlation (99% in clear days and 97% in cloudy days) between the irradiance measured in the solar resource monitorization array's horizontal pyranometer and the short-circuit current in the sensor's top solar pannel. This enables one to safely assume that short circuit current output the other faces of the cube can be directly correlated to the total irradiance on those faces, thus enabling us to substitute the pyranometers for solar panels, thus decreasing the cost associated to monitoring these variables.

Additionally to the construction of the sensor, an artificial neural network model was developed to calculate direct and diffuse irradiance from the data provided by the sensor. This phase of the project was completed using data from NREL. Several Networks were tested, with different sets of input and output data. The models were evaluated through RMSE values and used to produce estimates of solar irradiance (direct and difuse). These estimates were then compared to the real measurements performed at NREL with satisfactory results. The correlation between measured and predicted irradiance was again very high (80% or higher for cloudy days and 95 percent or higher for clear days).

Upon testing of the sensor's battery life, it was concluded that there is a serious need in furthering the study of microcontroller options and energy management sollutions. This accessment was made in the final sections of this work, where a new version of the sensor was devised. This new version of the sensor features great improvements in power consumption and requires a much smaller and cheaper battery to operate in the needed conditions. The new charging system will also improve the efficiency of the solar charging process greatly. Different configurations with less solar panels are also discussed in the latter section of the work, giving rise to cheaper versions of the sensor that rely more heavily on ANN models to estimate all the needed sunlight components.

8. References

Killinger, S., Braam, F., Müller, B., Wille-Haussmann, B., & McKenna, R. (2016). Projection of power generation between differently-oriented PV systems. *Solar Energy*, *136*, 153–165. https://doi.org/10.1016/j.solener.2016.06.075

Perez, R., Ineichen, P., Seals, R., Michalsky, J., & Stewart, R. (1990). Modeling daylight availability and irradiance components from direct and global irradiance. *Solar Energy*, *44*(5), 271–289. https://doi.org/10.1016/0038-092X(90)90055-H

Srikrishnan, V., Young, G. S., Witmer, L. T., & Brownson, J. R. S. (2015). Using multi-pyranometer arrays and neural networks to estimate direct normal irradiance. *Solar Energy*, *119*, 531–542. https://doi.org/10.1016/j.solener.2015.06.004

Texas Instruments Incorporated, May 1976, A7800 SERIES POSITIVE-VOLTAGE REGULATORS, Datasheet. http://www.ti.com/lit/ds/symlink/lm340.pdf

Nexperia B.V., 2017, 74HC4067; 74HCT406716-channel analog multiplexer/demultiplexer, product datasheet. https://assets.nexperia.com/documents/data-sheet/74HC_HCT4067.pdf

Microchip Technology Inc. ,2008, 2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface. https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf

9. Figure References

[1] – https://www.element14.com/community/docs/DOC-81294/l/raspberry-pi-3-model-b-with-1gb-of-ram-with-wifi-and-bluetooth-low-energy (Visited March 2018)

[2] - http://www.omniinstruments.co.uk/cm121-shadow-ring.html (Visited March 2018)

[3] - https://www.delta-t.co.uk/product/spn1/ (Visited March 2018)

[4] - https://consumer.huawei.com/en/mobile-broadband/ (Visited August 2018)

[5] - https://learn.sparkfun.com/tutorials/ (Visited June 2018)

[6] - https://learn.adafruit.com/thermistor/using-a-thermistor (Visited June 2018)

[7] - https://www.electrofun.pt/comunicacao/modulo-rtc-ds1307-relogio (Visited June 2018)

[8] - https://www.electrofun.pt/pesquisa?cat_id=all&controller=search&s=mpu+6050&spr_submit_search=Pesquisar&n=28 (Visited June 2018)

[9] - https://www.sparkfun.com/products/13906 (Visited June 2018)

[10] - http://electronic-studio.com/index.php?a=11025&lang=eng (Visited June 2018)

[11] - https://www.electronics-tutorials.ws/waveforms/555_oscillator.html (Visited June 2018)

[12] - https://electronics.stackexchange.com/questions/146936/op-amp-based-dc-dc-voltage-doubler (Visited June 2018)

[13] - https://vetco.net/products/single-relay-module-for-arduino (Visited June 2018)

[14] - https://www.google.com/maps/@38.7561368,-9.1562997,19z (Visited June 2018)

[15] - https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10424/104240B/Cloud-cover-forecast-from-a-ground-based-all-sky-infrared/10.1117/12.2278636.short?SSO=1 (Visited June 2018)

[16] - https://blog.goodaudience.com/artificial-neural-networks-explained-436fcf36e75 (Visited June 2018)

[17] - https://arduino.cc (Visited June 2018)

[18] - https://www.thethingsnetwork.org/map (Visited June 2018)

[19] - https://www.sigfox.com/en/coverage (Visited June 2018)

[20] - https://www.amazon.com/RioRand-LM2596-Converter-1-23V-30V-5Pcs-LM2596/dp/B008BHB4L8 (Visited June 2018)

10.Annexes

10.1. Annex 1 – python code

```
#--------------------------------LIBRARIES-------------------------------------
import datetime
import csv
import time
import math as math
import Adafruit_GPIO.SPI as SPI #SPI LIBRARY
import Adafruit_MCP3008        #MCP3008 LIBRARY
import RPi.GPIO as GPIO        #GPIO LIBRARY
import smbus                   #I2C LIBRARY
from Adafruit_IO import Client, Feed, RequestError #REST client

#--------------------------------VARIABLES-------------------------------------

ADAFRUIT_IO_KEY = '810deb121e304912a8a7e62efd49f43d'
ADAFRUIT_IO_USERNAME = 'exkerdino'
aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)

R=15000             #Series Resistor
B=3950              #B-parameter
T=25                #Ambient Temperature
DELTA=1             #sleep time in minutes
data=[0]*15
Delta=DELTA*60
MEASURE=1
CHARGE=0
power_mgmt_1 = 0x6b   # Power management registers
power_mgmt_2 = 0x6c   # Power management registers
PA=6
PB=13
PC=19
PD=26
MODE_SWITCH=5

#-------------------------------GPIO SETTINGS---------------------------------

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(PA,GPIO.OUT)
GPIO.setup(PB,GPIO.OUT)
GPIO.setup(PC,GPIO.OUT)
GPIO.setup(PD,GPIO.OUT)
GPIO.setup(MODE_SWITCH,GPIO.OUT)

#-----------------------------SPI SETTINGS--------------------------------------

CLK  = 11
MISO = 9
```

```
MOSI = 10
CS   = 8
mcp = Adafruit_MCP3008.MCP3008(clk=CLK, cs=CS, miso=MISO, mosi=MOSI)

#----------------------------I2C SETTINGS-------------------------------------

bus = smbus.SMBus(1) # or bus = smbus.SMBus(1) for Revision 2 boards
address = 0x69       # This is the address value read via the i2cdetect command
                # Now wake the 6050 up as it starts in sleep mode
bus.write_byte_data(address, power_mgmt_1, 0)

#-----------------------------FUNCTIONS-----------------------------------
   #-------------------------ACCELEROMETER

def read_byte(adr):
   return bus.read_byte_data(address, adr)

def read_word(adr):
   high = bus.read_byte_data(address, adr)
   low = bus.read_byte_data(address, adr+1)
   val = (high << 8) + low
   return val

def read_word_2c(adr):
   val = read_word(adr)
   if (val >= 0x8000):
      return -((65535 - val) + 1)
   else:
      return val

def dist(a,b):
   return math.sqrt((a*a)+(b*b))

def get_y_rotation(x,y,z):
   radians = math.atan2(x, dist(y,z))
   return -math.degrees(radians)

def get_x_rotation(x,y,z):
   radians = math.atan2(y, dist(x,z))
   return math.degrees(radians)

   #---------------------------TEMPERATURE-----------------------------------

def TEMP(i):

   t=298.15
   values =1023-mcp.read_adc(0)
   if values==1023:
      return
   if values==0:
      return
```

```
    r=float((15000*values)/(1023-values))

    steinhart = r/ 15000;
    steinhart = math.log(steinhart);
    steinhart /= B;
    steinhart += 1.0 / (t );
    steinhart = 1.0 / steinhart;
    steinhart -= 273.15;
    steinhart =round(steinhart,2)

    return steinhart
    #------------------------------CURRENT-------------------------------------

def CURR(i): #leitura da corrente

    values = mcp.read_adc(0)
    c= float(values )
    c*= 0.97751
    c/=*2           #corrente em mA
    c=round(c,2)

    return c

    #------------------------------MULTIPLEXER---------------------------------

def SC(i):
    GPIO.output(PA,GPIO.LOW)
    GPIO.output(PB,GPIO.LOW)
    GPIO.output(PC,GPIO.LOW)
    GPIO.output(PD,GPIO.LOW)
    x=[0]*4
    x=list('{0:4b}'.format(i))
    if x[0]=='1':
        GPIO.output(PA,GPIO.HIGH)
    if x[1]=='1':
        GPIO.output(PB,GPIO.HIGH)
    if x[2]=='1':
        GPIO.output(PC,GPIO.HIGH)
    if x[3]=='1':
        GPIO.output(PD,GPIO.HIGH)
    return
#--------------------------------MODE SWITCH---------------------------------

def switch(MODE):
    if (MODE==MEASURE):
        GPIO.output(MODE_SWITCH,GPIO.HIGH)
    if (MODE==CHARGE):
        GPIO.output(MODE_SWITCH,GPIO.LOW)
    return

#---------------------------------MAIN-----------------------------------------
```

```
   #-------------------------Start data feed----------------------------------

try: # if we have a 'analog' feed
   datafeed = aio.feeds('analog')
except RequestError: # create a analog feed
   feed = Feed(name='analog')
   datafeed = aio.create_feed(feed)

print("Started successfully") #DEBUG

   #-------------------------MAIN LOOP----------------------------------------

while(1):

      #SWITCH TO MEASURING STATE

   switch(MEASURE)
   print("MEASURING") #DEBUG

      #GET UNIX TIMESTAMP

   times=time.time()
   #print(times)      #DEBUG
   times=int(times)
   data[0]=times;

      #GET ACCELERATION
   aX = read_word_2c(0x3b)
   aY = read_word_2c(0x3d)
   aZ = read_word_2c(0x3f)

   aX_scaled = accel_xout / 16384.0
   aY_scaled = accel_yout / 16384.0
   aZ_scaled = accel_zout / 16384.0
   data[1]= round(get_x_rotation(accel_xout_scaled, accel_yout_scaled, accel_zout_scaled),2)
   data[2]= round(get_y_rotation(accel_xout_scaled, accel_yout_scaled, accel_zout_scaled),2)

      #GET CURRENTS AND TEMPERATURES
   for i in range(12):
      SC(i)
      if i<6:
         data[i+3]=TEMP(i)
      if i>5:
         data[i+3]=CURR(i)

      #PRINT TO CSV
   with open('/home/pi/EdgeServer/data'+ str(times) + '.csv', 'a') as csvfile:
      filewriter = csv.writer(csvfile, delimiter=',',quotechar='|',
                              quoting=csv.QUOTE_MINIMAL)
```

```
    filewriter.writerow(data)

with open('dados.csv', 'a') as csvfile:
    filewriter = csv.writer(csvfile, delimiter=',',quotechar='|',
                            quoting=csv.QUOTE_MINIMAL)

    filewriter.writerow(data)


print("Datapoint!")#DEBUGGING

    # SWITCH TO CHARGING STATE

switch(CHARGE)
print("CHARGING")

csvfile.close
aio.send(datafeed.key, data)

    # SLEEP for 1 minute
time.sleep(Delta)
```

10.2. Annex 2 - Shell script

```
sleep 20

cd umtskeeper

/home/pi/umtskeeper/umtskeeper --sakisoperators "USBINTERFACE='0' OTHER='USBMODEM'
USBMODEM='12d1:1c05' APN='CUSTOM_APN' CUSTOM_APN='CUBO'SIM_PIN='0000'
APN_USER='USER' APN_PASS='PWRD'" --sakisswitches "--sudo --console" --devicename 'Huawei' --log --
monthstart 8 --nat 'no' --httpserver &>> /home/pi/Desktop/error.log &

echo "STARTING MAIN"

sudo python -u /home/pi/Desktop/main.py
```